

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

факультет інформатики та обчислювальної техніки
(повна назва інституту/факультету)

кафедра автоматика та управління в технічних системах
(повна назва кафедри)

«На правах рукопису»
УДК _____

«До захисту допущено»

Завідувач кафедри

(підпис) О. І. РОЛІК
(ініціали, прізвище)

“ ” _____ 2018р.

Магістерська дисертація

зі спеціальності (спеціалізації) 126 «Інформаційні системи та технології»
(код і назва спеціальності)

на тему: Система обміну юридично значущими електронними документами між контрагентами

Виконав : студент 6 курсу, групи ІА-73мп
(шифр групи)

Головаш Максим Григорович
(прізвище, ім'я, по батькові) _____ (підпис)

Науковий керівник к.т.н., доцент Долина В.Г.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) _____ (підпис)

Консультант _____
(назва розділу) _____ (науковий ступінь, вчене звання, прізвище, ініціали) _____ (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) _____ (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2018 року

**Національний технічний університет України
“Київський політехнічний інститут
імені Ігоря Сікорського”**

Факультет інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматики та управління в технічних системах
(повна назва)

Ступінь вищої освіти – другий (магістерський)
(код, назва)

Спеціальність 126 «Інформаційні системи та технології»
(код, назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис) О. І. РОЛІК
(ініціали, прізвище)

“ ” _____ 2018_р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Головашу Максиму Григоровичу

(прізвище, ім'я, по батькові)

1. Тема дисертації Система обміну юридично значущими електронними документами між контрагентами

Науковий керівник дисертації Долина Віктор Георгійович Володимир Миколайович, к.т.н., доцент

затверджені наказом по університету від “ 29 ” жовтня 2018_р. № _____

2. Строк подання студентом дисертації “ 4 ” грудня 2018_р.

3. Об'єкт дослідження: процес обміну юридично значущими документами між контрагентами

4. Вихідні дані до проекту: технічне завдання

4.Зміст магістерської роботи: а) огляд і аналіз існуючих рішень; б) вибір та обґрунтування архітектури системи в) реалізація програмного комплексу; г) стартап-проект;

5. Перелік графічного (ілюстративного) матеріалу: Схема варіантів використання; Фнкціональна схема процесу надсилання документів; Структурна схема компонентів системи; Функціональна схема взаємодії з зовнішніми системами; Структурна схема бази даних; Схема розподілу функцій між клієнтом та сервером; Фнкціональна схема життєвого циклу веб сторінки; Креслення вигляду сторінки веб порталу.

6. Консультанти розділів дисертації:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання “ 29 ” жовтня 2018 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Огляд і аналіз існуючих рішень	02.10.2018	
2	Вибір архітектури реалізації	15.10.2018	
3	Досліди по реалізації	29.10.2018	
4	Реалізація	06.11.2018	
5	Стартап-проект	20.11.2018	
6	Оформлення текстової та графічної документації	2.12.2018	
7	Представлення до захисту	4.12.2018	

Студент

(підпис)

Головаш М.Г.

(ініціали, прізвище)

Керівник проекту

(підпис)

Долина В.Г.

(ініціали, прізвище)

АНОТАЦІЯ

Магістерська дисертація освітньо-кваліфікаційного рівня “магістр” на тему «Система обміну юридично значущими електронними документами між контрагентами». 107 с., 36 рис., 26 таб., 9 додатків, 31 джерело.

В роботі отримано нове вирішення актуальної задачі підвищення ефективності обміну документами між контрагентами через загальнодоступний портал з підтримкою ЕЦП.

Проаналізовано існуючі сервісні програми для захисту фінансових документів з використанням технології цифрового підпису. Розглянуто технологію цифрового підпису та існуючі стандарти.

Розроблений загальнодоступний портал обміну електронними документами з широким функціоналом. В розробці використовувались такі технології, як Asp.Net, Vue.js, ЕЦП.

Ключові слова: ЕЦП, Asp.Net, Vue.Js, хмарне сховище даних

SUMMARY

Master's dissertation of educational-qualifying level of "Master" on " System of exchange of legally significant electronic documents between contractors". 107 pgs., 36 fig., 25 tables, 9 applications, 31 sources.

In this paper we obtain a new solution of the important problem of increasing the efficiency of document exchange between business partners via a public portal with support for digital signature.

The existing service programs for the protection of financial documents using digital signature technology are analyzed. The technology of digital signature and existing standards are considered.

A public portal for the exchange of electronic documents with a wide range of functions has been developed. The development used technologies such as Asp.Net, Vue.js, Digital signature.

Keywords: Digital signature, Asp.Net, Vue.Js, cloud storage

ЗМІСТ

ВСТУП.....	6
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	8
2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	11
2.1 Огляд програмного комплексу М.Е.Дос.....	11
2.2 Огляд програмного комплексу «Птах».....	14
3 ВИБІР ТА ОБҐРУНТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ.....	16
3.1 Сервери баз даних	18
3.2 Сервери додатків	18
3.3 Варіанти доступу клієнтів	19
3.4 Процеси серверів додатків	21
3.5 Потоки даних	23
3.5.1 Зв'язок серверів бази даних і серверів додатків, web-серверів.....	23
3.5.2 Зв'язок сервера і клієнта	23
3.5.3 Розподіл функцій між клієнтом і сервером	24
3.5.4 Сеанс роботи клієнта	24
3.6 Варіанти конфігурацій серверів додатків.....	25
3.6.1 Один сервер додатків, один web-сервер	26
3.7 Розподілення навантаження.....	27
3.8 Синхронізація ПО між серверами додатків.....	28
3.9 Використання робочої станції як локального сервера додатків	28
4 ФОРМУВАННЯ МАТЕМАТИЧНОЇ МОДЕЛІ ТА ЗАГАЛЬНОЇ СТРУКТУРИ ПРОГРАМНОГО КОМПЛЕКСУ	38
4.4.1 Прив'язки.....	44

4.4.2 Види вкладень.....	47
4.4.3 Статуси видів вкладень.....	53
4.4.4 Налаштування доступу до вкладень.....	54
4.4.5 Шаблони вкладень	56
4.4.6 Розділи документів.....	58
4.4.7 Налаштування доступу до управління контентом.....	58
4.4.8 Режими роботи з вкладеннями.....	59
4.5 ЕЦП в додатках.....	64
4.6 Реалізація управління користувачами.....	68
4.7.1 Web-розрахунки	74
4.7.2 Виклик web-сервісів.....	77
4.7.3 Формат обміну даними.....	78
4.7.4 Основні принципи взаємодії з web-сервісами з клієнтського додатка.....	78
5 ТЕСТУВАННЯ ПРОГРАМНОГО КОМПЛЕКСУ.....	84
5.1 Класифікація видів тестування програмного забезпечення.....	84
5.2 Тестування основних функціональних можливостей.....	86
5.3 Реалізація автоматичного тестування.....	88
6 СТАРТАП-ПРОЕКТ.....	90
6.1 Опис ідеї проекту (товару, послуги, технології).....	90
6.2 Технологічний аудит ідеї проекту.....	91
6.3 Аналіз ринкових можливостей запуску стартап-проекту.....	92
6.4 Розроблення ринкової стратегії проекту.....	98
6.5 Розроблення маркетингової програми стартап-проекту.....	100
6.5 Висновки.....	103
ВИСНОВКИ.....	104

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	105
---------------------------------	-----

ДОДАТОК А

ДОДАТОК Б

ДОДАТОК В

ДОДАТОК Г

ДОДАТОК Д

ДОДАТОК Е

ДОДАТОК Ж

ДОДАТОК З

ДОДАТОК К

ВСТУП

Розвиток людської цивілізації супроводжується вражаючим уяву збільшенням обсягу створюваної, оброблюваної і збереженої інформації. Збільшення обсягів інформації та зміни попиту на інформацію стали пред'являти нові вимоги до організації інформаційно документаційного обслуговування на підприємстві.

Управлінська діяльність і функціонування будь-якого підприємства в Україні, як і у всіх розвинених країнах, здійснюється за допомогою документів, які одночасно є джерелом, результатом і інструментом цієї діяльності.

При збільшенні обсягів інформації все більше очевидна неефективність використання паперових документів. Ручний пошук і обробка документів гальмують роботу організації. Крім цього у паперового документообігу є такі недоліки, як тривала підготовка погоджень, а значить і низька швидкість обробки інформації та реакції на нові впливи; низька швидкість передачі інформації особам, які приймають рішення. Більш того, використання паперового документообігу вимагало істотних матеріальних, кадрових витрат і значних тимчасових витрат на його обробку. З роками і збільшенням накопичених документів на підприємстві, даний процес все більше і більше посилювався.

В умовах сучасної економіки паперовий документообіг виявляється не в змозі забезпечити швидко і ефективно обробку великих обсягів інформації, від якої залежить успіх будь-якого підприємства. Таким чином, паперовий документообіг стає великою перешкодою на шляху ефективної роботи підприємства.

Розвиток глобальних комунікацій в діловому і повсякденному житті привів до появи нової області взаємовідносин, предметом яких є електронний обмін даними. У такому обміні даними можуть брати участь органи державної влади, комерційні і некомерційні організації, а також громадяни в своїх офіційних і особистих стосунках. Проблема збереження електронних документів від копіювання, модифікації і підробки вимагає для свого вирішення специфічних

засобів і методів захисту. Одним з поширених в світі засобів такого захисту є Електронний цифровий підпис (ЕЦП), який за допомогою спеціального програмного забезпечення підтверджує достовірність інформації документу, його реквізитів і факту підписання конкретною особою.

Наразі система IT-Enterprise не надає можливості організувати електронний обмін документами так, щоб документи при цьому залишалися юридично значимим. Обов'язковим атрибутом юридично значимого електронного документу є електронний цифровий підпис (ЕЦП).

На ринку інформаційних технологій є чимало рішень з електронного документообігу. Загальний недолік таких рішень – їх важко впровадити у іншу систему, оскільки необхідно реалізовувати складний функціонал перетворення даних із однієї системи у іншу.

Термінальні рішення вимагають установки на комп'ютер і постійного оновлення, які теж потрібно завантажувати і встановлювати.

Хмарні сервіси, розміщуються на хмарних серверах і працюють через браузер, тобто установки на ПК не вимагають. У них відсутні будь-які вимоги до пристроїв, їх потужності та конфігурації. Щоб використати досить просто зайти на сайт. Оновлення хмарних продуктів відбувається автоматично, і остання версія відразу стає доступною всім користувачам.

Таким чином, можна зробити висновок, що актуальність теми, якою ми вирішили присвятити дану роботу, підказана нам і обумовлена самим життям, тими економічними і соціальними умовами, які склалися в даний час у сфері бізнес.

Метою роботи є розробка системи обміну юридично значущими електронними документами між контрагентами.

Предметом досліджень є програмний комплекс «Система обміну юридично значущими електронними документами між контрагентами».

Об'єктом досліджень є процес оптимізації якості, витрат та часу на обмін документами між контрагентами.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

В ринковій економіці інформація стала товаром і її призначення, використання і зберігання підпорядковується законам ринку. Вміння грамотно і ефективно використовувати інформацію визначає швидкість розвитку підприємства, його ринкову вартість, здатність конкурувати з іншими підприємствами.

Часто робота з документами на підприємстві займає у співробітників велику частина часу, відволікаючи від вирішення нагальних питань і завдань. Це пов'язано з невміло побудованою системою управління документами, або її відсутністю.

Організація документообігу не просто економить трудовитрати, а й сприяє здійсненню більш точного контролю з боку керівництва за станом справ на підприємстві. Документи, необхідні для управління не просто цінні самі по собі. Принцип документування діяльності полягає в тому, що вся необхідна інформація повинна бути відображена в документах, документи повинні зберігатися так, щоб вони не були загублені, їх легко можна було знайти. Крім того, робота з документами повинна бути побудована так, щоб сприяти основній діяльності підприємства. Вибудована система документообігу повинна забезпечити такий стан роботи з документами, при якому кожен співробітник може досить швидко отримати і обробити необхідні документи і потрібну інформацію.

При паперовому документообігу цього досягти стає досить складно. З цими проблемами допомагає впоратися автоматизація процесів документообігу, яка допомагає збільшити швидкість роботи, скоротити втрати, пов'язані з інформаційним обсягом, підвищити якість системи управління в цілому.

Найбільш важливим стримуючим фактором розвитку електронної комерції є відсутність документального оформлення угод, проведених за допомогою Інтернету. Дана ситуація робить актуальною проблему оформлення і використання електронного документа (документа в електронній формі). Для виконання електронним документом функцій, що покладаються на нього, використовується електронний цифровий підпис (ЕЦП). ЕЦП за допомогою спеціального

програмного забезпечення підтверджує достовірність інформації документу, його реквізитів і факту підписання конкретного документу конкретною особою.

В даний час ЕЦП широко використовується. У діяльності підприємства, а також в особистих цілях електронний підпис виконує функції, аналогічні до підпису людини.

Електронний підпис є сукупність електронних даних, за допомогою якої можна легко і надійно упевнитися в тому, що: саме цей користувач згенерував дану електронний підпис. Жоден інший користувач не може згенерувати таку ж електронний підпис; саме цей документ або набір документів підписав користувач. Документ не був ніяк змінений або замінений після накладення підпису

Одним з головних пріоритетів України, визначених Законом України «Про основні засади розвитку інформаційного суспільства в Україні на 2007-2015 роки», є прагнення побудувати орієнтовне на інтереси людей, відкрите для всіх і спрямоване на розвиток інформаційне суспільство, в якому кожен міг би створювати і накопичувати інформацію та знання, мати до них вільний доступ, користуватися і обмінюватися ними, сприяючи суспільному і особистому розвитку та підвищуючи якість життя [3].

Сучасний правовий статус безпаперового документообігу із застосуванням електронного цифрового підпису визначається двома законами України, прийнятими у 2003 році: «Про електронні документи та електронний документообіг» (№ 851-IV) [1] і «Про електронний цифровий підпис» (№ 852-IV) [2].

Постійне збільшення кількості інформації, необхідної для прийняття адекватних управлінських рішень, призводить до того, що традиційні методи роботи з документами стають все більше нерентабельними. Так, за статистичними даними, 15% паперових документів губляться, а для їх пошуку працівники витрачають близько 30% свого часу.

Система обміну електронним документами допоможе вирішити данні проблеми. Наприклад, замість того, щоб кур'єром доставляти в банк паперовий

документ з підписами і печаткою достатньо надіслати по відкритих каналах електронний документ, завірений електронним підписом (або декількома підписами). Наявність електронного підпису гарантує, що документ створений саме цими людьми, отже, він рівнозначний аналогічному паперовому документу.

2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

2.1 Огляд програмного комплексу М.Е.Дос

«М.Е.Дос» (My Electronic Document, також Medoc, Медок) — поширене українське програмне забезпечення для подання звітності до контролюючих органів та обміну юридично значущими первинними документами між контрагентами в електронному вигляді.

Програма М.Е.Дос розроблена українськими програмістами під керівництвом Олесі Линник, яка в 2000 році продовжила сімейну справу свого батька Сергія Линника, що займається розробкою бухгалтерських систем з 1990 року і заснував групу компаній «Інтелект-Сервіс». Вже в 1993 році його розробка «БЕСТ ЗВІТ» була першою в Україні програмою для автоматизації роботи малого і середнього бізнесу. «БЕСТ ЗВІТ» стала праматір'ю сучасної програми М.Е.Дос, якою на сьогоднішній день користується переважна більшість компаній в країні.

З швидким розвитком нових технологій і вимог ринку програмне забезпечення безперервно актуалізувалося і вдосконалювалося, результатом чого в 2010 році стала поява на українському ринку передової розробки — М.Е.Дос.

У 2013 році в цій програмі об'єднали напрацювання з різних напрямків автоматизації бухгалтерської діяльності: звітності, розрахунку заробітної плати, взаємодії всередині корпорацій і обміну документами між контрагентами. Таким чином, М.Е.Дос став багатофункціональним комплексом з максимальними можливостями для бухгалтерів в країні. Програмне забезпечення задовольняє потреби в роботі з електронними документами компаній будь-якого масштабу, форми власності та виду діяльності.

Розробки «Інтелект-Сервіс» дозволили перевести процес подачі звітності в Україні в електронну форму. Спочатку ця процедура відбувалася з використанням дискет, а з часом стала повністю дистанційною з використанням електронного

цифрового підпису (ЕЦП). Зокрема, програма М.Е.Дос стала родоначальником електронного документообігу в країні.

М.Е.Дос розповсюджується і обслуговується через широку партнерську мережу, яка налічує понад 600 представництв по всій Україні: дилерів, посередників, дистриб'юторів і франчайзі.

В результаті розвитку і трансформації групи компаній «Інтелект-Сервіс» на початку 2017 року утворилася компанія Linkos group, яка акумулює в собі високотехнологічні програмні рішення для бізнесу і бюджетних установ.

Програма М.Е.Дос призначена для подачі звітності в усі контролюючі органи України (ДФСУ, ДССУ, ПФУ, ФСС з ТВП, ДКСУ, міністерства і відомства), для реєстрації податкових накладних та юридично значущого електронного документообігу. Крім того, програма надає модулі для нарахування зарплати, звітності великих компаній з розгалуженою структурою підрозділів і роботи банків.

М.Е.Дос містить всі актуальні форми звітності, які оперативно оновлюються відповідно до законодавства. Програма підтримує роботу з ЕЦП найбільш використовуваних центрів сертифікації, а також захищеними носіями для ЕЦП SecureToken.

Над програмою працює численний штат розробників, аналітиків, тестувальників, фахівців техпідтримки та інших напрямків. М.Е.Дос отримав позитивний експертний висновок у сфері технічного захисту інформації рівня ГЗ від Держспецзв'язку, що свідчить про високий рівень безпеки і захищеності програми.

У комп'ютерній програмі М.Е.Дос можна використовувати сертифікати ЕЦП, отримані в наступних акредитованих центрах сертифікації ключів (АЦСК):

- АЦСК «Україна» (Центр сертифікації ключів «Україна»);
- АЦСК ІДЦ ДФС (Інформаційно-довідковий центр ДФС);
- АЦСК Укрзалізниця;

- АЦСК Держінформюст;
- АЦСК Укрсиббанк;
- АЦСК «Masterkey»;
- АЦСК ТОВ НВФ «УНІС» (Українські національні інформаційні системи);
- АЦСК УСС (Українські спеціальні системи);
- АЦСК МВС України;
- АЦСК НБУ;
- АЦСК Збройних сил;
- АЦСК Приватбанк.

Програма підтримує роботу з захищеними носіями для ЕЦП. У Програмі М.Е.Дос також реалізовано автоматичне продовження сертифікатів цифрових підписів.

Розробник програми надає код доступу кожному підприємству. Якщо в програму не введений код доступу, тоді вона буде працювати в демонстраційній версії, яка дозволяє ознайомитися з функціональними можливостями програми, але функції експорту, відправки електронною поштою і друк документів будуть недоступні.

Програма має як готові рішення для основних потреб користувачів, так і додаткові модулі.

Рішення М.Е.Дос.Держава — подача всіх видів звітності в усі контролюючі органи, реєстрація податкових накладних в Єдиному реєстрі податкових накладних та обмін ними з контрагентами.

Рішення М.Е.Дос.Бізнес — обмін первинними бухгалтерськими документами з контрагентами.

Модуль М.Е.Дос.Акциз та ТТН — для роботи з системою електронного адміністрування реалізації палива та поводження з товарно-транспортними накладними.

Модуль М.Е.Дос.Зарплата — розрахунок і нарахування заробітної плати співробітникам, облік і управління персоналом.

Модуль М.Е.Дос.Корпорація — консолідація звітності підприємств з розгалуженою структурою.

Недоліком даної системи є те, що термінальні рішення вимагають установки на комп'ютер і постійного оновлення, які теж потрібно завантажувати і встановлювати.

2.2 Огляд програмного комплексу «Птах»

Платформа «Птах» має такі можливості:

- обмін більше 1 000 000 документами за день;
- робота з будь-якими типами документів;
- інтеграція з будь-якою обліковою системою через API;
- пряма інтеграція в 1С:Підприємство Flydoc;
- обмін первинними документами через Медок, Сота, FREDO ДокМен;
- отримання документів на електронну пошту або через веб;
- інтеграція frame у власний сайт для реалізації онлайн документообігу.

При обміні документів реалізований стандарт передачі з використанням так званого конверту, в який запаковується: XML - файл, де знаходиться вся інформація в електронному вигляді для завантаження в облікову систему, PDF - файл візуально представлений документ (сканкопія документу) та вкладений файл — довільний файл в разі необхідності.

Кожен файл підписаний ЕЦП та зашифрований на сертифікат отримувача, що унеможливорює зміну чи перегляд документу третіми особами.

В платформі «Птах» реалізовано єдиний довідник шаблонів електронних документів з наступними атрибутами: обов'язкові контролі, комплект підпису для

документів, маршрут документу (односторонній - рахунок, двосторонній - акт, накладна), можливість замовлення паперового документу; єдиний довідник контрагентів з автоматичною звіркою з ЄДК.

Система сповіщення дозволяє: налаштувати для кожного отримувача чи документу окремий варіант сповіщення, повідомляти клієнтів через sms, viber, telegram, а також моментального відкриття електронних документів в Web, Mobile, налаштувати регулярність сповіщення та кінцевий термін узгодження документу.

Отже, програмний комплекс «Птах» — це відкрита платформа, що об'єднує всіх учасників документообміну, забезпечує єдиний стандарт обміну, перевірки, захисту та транспортування юридично значущих документів, підписаних ЕЦП.

Недоліками денного рішення є обмежений функціонал та недостатньо розвинене зовнішнє арі.

3 ВИБІР ТА ОБҐРУНТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ

Для реалізації даного рішення було обрано сервісно-орієнтовану архітектуру.

Сервісно-орієнтоване програмування (або архітектура) — це подальший розвиток компонентного підходу до розробки компонентних програмних систем (ПС), заснований на використанні програм-сервісів із стандартизованими інтерфейсами.

Компоненти ПС можуть бути розподілені по різних вузлах мережі, і пропонуватися як незалежні, слабо зв'язані, замінювані сервіси-застосування. Інтерфейс компонентів ПС забезпечує інкапсуляцію деталей реалізації кожного конкретного компоненту від інших. Таким чином, сервісно-орієнтована архітектура надає гнучкий спосіб комбінування і повторного використання компонентів для побудови складних розподілених програмних систем, зокрема, корпоративних програмних систем. SOA створює комунікаційне середовище для модулів, що реалізують прикладну бізнес-логіку. Інформація про модулі публікується в такій формі, що їх використання не вимагає знань про використані в них рішення і технології.

До складу програмного комплексу входять наступні компоненти:

- веб додаток — клієнт для користувачів;
- центральний сервіс - керуючий комплект ERP системи з встановленим модулем «Система управління сервісами»;
- фінансовий сервіс — зовнішній програмний продукт, який використовується для фінансового обліку та формування фінансових документів і як джерело інформації про надходження оплати;
- сервіс обміну документами — комплект ERP системи з встановленим модулем «Обмін документами».

В свою чергу, комплект ERP системи має трирівневу архітектуру.

Виділяється три рівня виконання програми (рисунок 3.1):

- рівень баз даних;

- рівень серверів додатків;
- рівень клієнта.

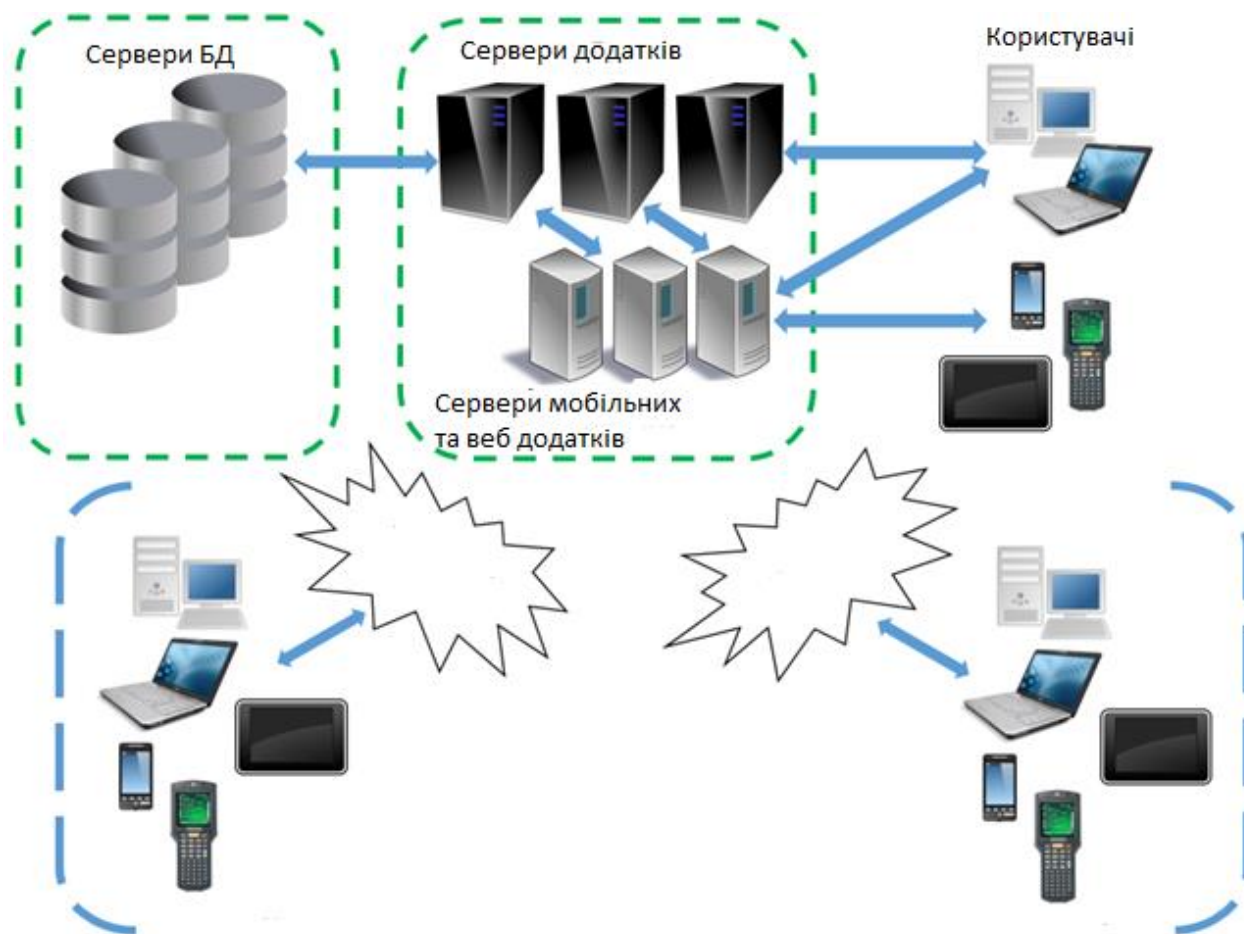


Рисунок 3.1 – Трирівнева архітектура

З даними працює ряд серверів баз даних. В якості серверів баз даних використовуються Microsoft SQL Server 2014/2016/2017 або Oracle Database 11/12. У найпростішому випадку сервер баз даних один. Для масштабування кількість використовуваних серверів баз даних не обмежена.

В серверах додатків виконується вся бізнес-логіка обробки даних. Сервер додатків формує запити до серверу бази даних, виконує розрахунки і пересилає готові результати. Кількість серверів додатків не обмежена, що забезпечує масштабованість системи.

Для web-доступу, а також роботи мобільних додатків застосовуються web-

сервери (сервери мобільних і веб-додатків). Дані сервери служать для взаємодії з клієнтом (браузером) і мобільними додатками. Кількість web-серверів не обмежена, що забезпечує масштабованість системи.

Windows-клієнт — клієнтське ПЗ розташоване на кожній кінцевій клієнтській робочій станції. Клієнтське ПЗ обмінюється даними з одним із серверів додатків.

Web-клієнт — web-браузер обмінюється даними з web-сервером.

Мобільні додатки — мобільний додаток обмінюється даними з web-сервером.

3.1 Сервери баз даних

Сервери баз даних (БД) розташовуються в захищеній зоні мережі. Доступ до серверів БД мають тільки сервери додатків і web-сервери. Зазвичай, для безпеки сервери БД навіть не мають фізичного зв'язку із зовнішнім світом.

Обмін даними з серверами додатків відбувається по протоколу TCP/IP, сервер додатків встановлює з'єднання (connect) з серверами БД, посилає запити sql, отримує у відповідь дані.

Дані в різних серверах баз даних можуть бути сегментовані (технологія розділення бази на частини по різних серверів), зеркальовані (технологія підтримки дзеркальних копій для паралельного доступу), розбиті по різних сервісів (технологія підтримки єдиних даних різних комплектів) і т. п. Всі ці технології вкупі дають можливість необмеженого масштабування на рівні БД.

3.2 Сервери додатків

На серверах додатків зосереджена і виконується вся бізнес-логіка програмного забезпечення.

Серверів додатків може бути як завгодно багато. Підвищення кількості серверів додатків збільшує обчислювальну потужність системи IT-Enterprise в цілому. Передбачено динамічне балансування навантаження по серверам.

Сервери додатків також виконують функцію підтримки єдиної версії ПЗ на всіх рівнях: на самих серверах додатків автоматично підтримується єдина версія, яка також синхронізована:

- зі схемою даних на сервері БД з одного боку;
- з клієнтським ПЗ з іншого боку.

Для web-доступу клієнтів призначений пул web-серверів додатків. На web-серверах зосереджена бізнес-логіка web-додатків, комунікаційні засоби прийому/передачі запитів для забезпечення роботи веб-сервісів і мобільних додатків.

Web-серверів може бути як завгодно багато. Підвищення кількості web-серверів збільшує обчислювальну потужність системи в цілому. Передбачено динамічне балансування навантаження по web-серверам.

3.3 Варіанти доступу клієнтів

Для роботи з IT-Enterprise підтримуються кілька варіантів доступу для клієнтів.

Тонкий Windows-клієнт — windows-додаток, який приєднується до сервера додатків, посилає запити до сервера додатків, ініціює виконання розрахунків на сервері додатків, отримує підсумкові набори даних, що відображає дані клієнта, реалізує користувальницький інтерфейс. Працює по протоколах tcp або http. Програмне забезпечення встановлюється на клієнтському комп'ютері. Клієнт реалізує інтерфейс користувача і взаємодію з кінцевими пристроями.

Web-клієнт «IT.Web 2016» (IT-Enterprise Web 2016) — працює в web-браузері. Оптимізовано для роботи на персональних комп'ютерах. Це веб-додаток, який через web-сервери в результаті приєднується до сервера додатків, посилає

запити до сервера додатків, ініціює виконання розрахунків на сервері додатків, отримує підсумкові набори даних, що відображає дані клієнта, реалізує користувацький інтерфейс. Працює по протоколах http/https. Оптимізовано для використання на традиційному моніторі, роботи за допомогою клавіатури і миші. Підтримує роботу на мобільних пристроях (планшет, смартфон). Працює на будь-яких операційних системах.

Web-клієнт «IT.Web RMD» (IT-Enterprise Responsive & Material Design (by Google)) — працює в web-браузері. Оптимізовано для роботи на будь-яких, в тому числі мобільних пристроях (десктоп, планшет, смартфон). Самостійно адаптується під пристрій, підтримуючи постійну читабельність інформації. Це веб-додаток, який через web-сервери в результаті приєднується до сервера додатків, посилає запити до сервера додатків, ініціює виконання розрахунків на сервері додатків, отримує підсумкові набори даних, що відображає дані клієнта, реалізує користувацький інтерфейс. Працює по протоколах http/https. Працює на будь-яких операційних системах.

Робота за допомогою web-додатків і web-віджетів — web-додатки, які реалізують обмежений набір завдань з усієї системи IT-Enterprise. Забезпечують зручний користувацький інтерфейс, всю бізнес-логіку виконують сервери додатків. Клієнт працює через браузер. Додатки можуть вбудовуватися в інші сайти і портали. Працює по протоколах http/https.

Робота за допомогою мобільних додатків — мобільні додатки для платформ Android, iOS, Windows Phone, які реалізують обмежений набір завдань з усієї системи IT-Enterprise. Забезпечують зручний користувацький інтерфейс, всю бізнес-логіку виконують сервери додатків. Клієнт встановлює додаток з магазину на свій мобільний пристрій. Працює по протоколах http, https (для деяких мобільних ОС можлива робота тільки по https).

Web-сервіси — можливості для роботи з IT-Enterprise програмного забезпечення 3-ї сторони. Надається стандартний інтерфейс взаємодії, розмежування прав доступу і т. п. Працює по протоколах http/https.

3.4 Процеси серверів додатків

Сервер додатків IT-Enterprise складається з наступних процесів: слухач, сесія клієнта, віртуальна машина, воркери, сервери мобільних та веб додатків.

Слухач (Launcher) — процес, який спілкується з web-додатками і клієнтськими додатками: приймає запити, пересилає готові результати розрахунків, управляє сесансами. Даний компонент є сполучною ланкою між web-додатками, клієнтськими додатками та іншими внутрішніми процесами. Реалізація: Microsoft .Net, мова C#. Розрядність: 64 або 32 (в залежності від розрядності операційної системи).

Сесія клієнта (SessionHost) — запускається на сервері додатків для кожного клієнтського додатка і виконує бізнес-логіку сесансу, взаємодіє з сервером БД (рисуюнок 3.2). Реалізація: Microsoft .Net, мова C#. Розрядність: 64 або 32 (в залежності від розрядності операційної системи).



Рисуюнок 3.2 – Процеси на сервері додатків

Віртуальна машина VFP (ArmHost) — може запускатися для кожного клієнта, служить для виконання legacy VFP-коду, взаємодіє з сервером БД. Реалізація: Microsoft .Net, мова C#, використовує dll-бібліотеку, реалізовану в Visual FoxPro. Розрядність: тільки 32.

Воркеры (worker) — обробка разових запитів web-додатків, мобільних додатків, web-сервісів. На відміну від сесій клієнтів не зберігає контекст сесії, існує

завжди — не породжується при запиті, не знищується після обробки запиту. Реалізація: виділені сесії SessionHost. Кількість воркерів динамічно змінюється в залежності від навантаження, що надходить.

Додатково в пулі серверів додатків виділяються сервери мобільних і web додатків для роботи мобільних і веб-додатків використовується додаткова служба, яка є проміжною ланкою між клієнтськими мобільними і web програмами та серверами додатків IT-Enterprise. Всі виклики від мобільних клієнтських додатків і web додатків приймаються сервером мобільних і web додатків, передаються на обробку серверам додатків. Результати розрахунків надходять у зворотному порядку і відсилаються клієнту. Тобто сам сервер не виконує якихось розрахунків, а служить передавальною ланкою і реалізує користувацький інтерфейс. Реалізація: Microsoft .Net, ASPX, мова C#, JS. Розрядність: 64 або 32 (в залежності від розрядності операційної системи)

У кожен момент часу активний тільки один з серверних процесів клієнта SessionHost або ArmHost (рисунок 3.3).

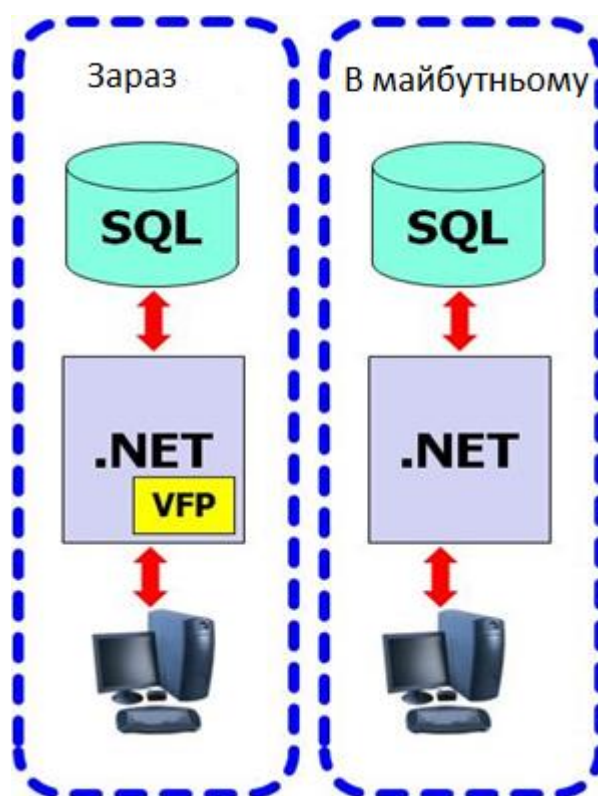


Рисунок 3.3 – Серверні процеси SessionHost та ArmHost

Управління між процесами передається в залежності від коду, який необхідно виконати. Для оптимізації роботи процес ArmHost запускається завжди і в режимі очікування очікує передачі йому управління. При необхідності виконання VFP-коду управління передається у віртуальну машину VFP (ArmHost), ArmHost виконує цей код, після чого керування знову передається назад головному ядру IT-Enterprise. В майбутніх версіях IT-Enterprise віртуальна машини і VFP-код підтримуватися не будуть.

3.5 Потоки даних

3.5.1 Зв'язок серверів бази даних і серверів додатків, web-серверів

Для оптимізації роботи рекомендується розміщувати всі сервери баз даних, сервери додатків і web-сервери в одній локальній обчислювальній мережі. Або в географічно рознесених дата-центрах з каналом зв'язку з гарантованою якістю зв'язку.

Обмін даними між сервером бази даних і серверами додатків відбувається за допомогою SQL-запитів. Сервер додатків посилає SQL-запити серверу бази даних. Сервер бази даних виконує обробку даних і повертає набори даних серверу додатків. Обмін даними між серверами додатків і web-серверами відбувається по протоколу tcp. Рекомендується зв'язок між серверами по гігабітному каналу Ethernet (або вище).

3.5.2 Зв'язок сервера і клієнта

Сервер додатків і клієнтське робоче місце повинні бути пов'язані по мережі. Аналогічно, web-сервер і клієнт повинні бути пов'язані по мережі. Це може бути:

- локальна обчислювальна мережа;
- віртуальна приватна мережа (VPN-канал);
- зв'язок через Інтернет;
- або будь-який інший канал зв'язку.

Необхідно забезпечити достатню швидкість обміну даними між клієнтом і сервером додатків (див. в документі «Вимоги ...» пункт «Вимоги до мережі»). Обмін даними між сервером додатків і клієнтським робочим місцем оптимізований для нормальної швидкості роботи системи навіть на слабких каналах зв'язку.

3.5.3 Розподіл функцій між клієнтом і сервером

Сервер додатків виконує всі функціональні дії і розрахунки. Клієнтське робоче місце забезпечує тільки введення/виведення даних у користувача (рисунок 3.4):

- відображає результати розрахунків на екрані;
- виконує друк сформованих сервером додатків звітів на принтер
- виконує введення даних
- пересилання введеної інформації серверу додатків і т.п.



Рисунок 3.4 – Розподіл функцій між клієнтом і сервером

3.5.4 Сеанс роботи клієнта

При запуску повнофункціонального клієнтського ПЗ (windows-клієнт, web-клієнт IT.Web2016 або web-клієнт IT.WebRMD) створюється новий сеанс роботи клієнтської робочої станції з сервером додатків. Управління процесом виконує сервер додатків.

Сервер додатків ініціює появу чергового діалогу у клієнта. Разом з командою «Вивести діалог» передаються параметри діалогу і дані. Після завершення діалогу, введені дані від клієнта пересилаються на сервер додатків, який продовжує роботу.

У windows-клієнта програмні модулі відображення діалогів реалізовані як стандартні функції клієнта і розміщені на клієнті. Від сервера додатків клієнту пересилаються тільки дані, все навантаження по інтерфейсу покладено на клієнта.

У web-клієнті IT.Web2016 програмні модулі відображення діалогів реалізовані як ASP код на web-сервері. Від web-сервера клієнту пересилається готовий HTML, все навантаження по інтерфейсу покладено на сервер.

У web-клієнті IT.WebRMD програмні модулі відображення діалогів реалізовані як JS-код на клієнті. Від web-сервера клієнту пересилаються тільки дані. JS-код кешується і не бере участі в постійній пересиланні. Навантаження по інтерфейсу розподілена між сервером і клієнтом, більша частина покладена на клієнта.

3.6 Варіанти конфігурацій серверів додатків

Сервер додатків – це програма (мережевий сервіс), яка працює на одному з комп'ютерів. Слід розрізняти поняття «сервер» як апаратний пристрій і «сервер додатка» як програму, що працює на будь-якому комп'ютері.

Кількість серверів додатків не обмежена. Web-сервер – це теж програма (мережевий сервіс). Кількість web-серверів не обмежена.

Виходячи з того, що «сервер» - це програма, а також, що серверів може бути кілька, можна виділити кілька варіантів розміщення серверів додатків і web-серверів:

- один сервер додатків, розміщений на окремому комп'ютері;
- один сервер додатків, поєднаний з сервером баз даних;
- web-сервер на окремому комп'ютері;
- web-сервер, поєднаний з сервером додатків;

- web-сервер, поєднаний з сервером додатків і сервером бд;
- кілька виділених серверів додатків;
- кілька виділених web-серверів;
- робоча станція – локальний сервер додатків;
- всілякі поєднання різних варіантів.

3.6.1 Один сервер додатків, один web-сервер

В даному варіанті виділяється один сервер, на якому розміщується програма «сервер додатків» і, можливо, web-сервер». Всі клієнтські робочі станції використовують один сервер додатків / один web-сервер.

У разі невеликої кількості робочих місць (5-10) програми «сервер додатків» і «web-сервер» можуть бути встановлені безпосередньо на сервер бази даних. Таким чином один апаратний або віртуальний комп'ютер може бути використаний для двох або трьох серверних програм – сервера бази даних, сервера додатків, web-сервера.

Переваги та недоліки: єдине місце розміщення програм серверів додатків/web; простота адміністрування обслуговування і оновлення; високі вимоги до апаратного забезпечення при підвищенні кількості клієнтських підключень; надійність роботи системи залежить від двох/трьох серверів – сервера бази даних, сервера додатків, web-сервера, при виході з ладу будь-якого з них робота системи неможлива

Потужність сервера визначається технічними вимогами, описаними в документації «Вимоги ...». Якщо для забезпечення всіх клієнтських підключень необхідний більш потужний сервер, навантаження може бути розподілена на кілька фізичних або віртуальних серверів.

3.6.2 Кілька серверів додатків, кілька web-серверів

В даному варіанті виділяється кілька серверів додатків і кілька web-серверів, які обслуговують усі клієнтські підключення. Даний варіант є оптимальним для

використання (варіант «один сервер програми і один web-сервер» є окремим випадком даного варіанту роботи).

Переваги та недоліки: для забезпечення необхідних обчислювальних потужностей достатньо виділити кілька серверів середнього класу замість одного сервера високого класу, що в загальному випадку дешевше; з'являються додаткові роботи по розподілу навантаження між серверами та синхронізації програмного забезпечення між серверами; підвищується надійність роботи, якщо один з серверів вийшов з ладу, це не тягне за собою припинення роботи системи, всього лише буде підвищено навантаження на залишилися сервери додатків / web; масштабованість системи. Підвищення обчислювальної потужності досягається установкою нового сервера в ряд серверів додатків/web або заміною одного сервера на інший, більш потужний.

3.7 Розподілення навантаження

Розподіл клієнтів по серверам може бути статичний, динамічний або змішаний.

У разі статичного розподілу всі клієнти поділяються на групи. Кожна група використовує окремий сервер додатків або web-сервер, закріплений за даною групою.

У разі динамічного розподілу клієнт звертається із запитом на «центральный сервер додатка» / «центральный web-сервер, який виконує перенаправлення запиту на найменш завантажений в даний момент сервер додатка / web-сервер. Алгоритми визначення завантаження серверів додатків можуть бути різні.

Змішаний розподіл — можна поєднувати зазначені два види розподілу навантаження. Для частини користувачів виділяється окремий сервер (або кілька), які обслуговують тільки користувачів даної групи. Всі інші користувачі рівномірно розподіляються на залишився пул серверів.

Переваги та недоліки: при динамічному розподілі досягається рівномірне завантаження всіх серверів; при статичному розподілі досягається резервування потужностей під потреби конкретних клієнтських станцій.

3.8 Синхронізація ПО між серверами додатків

Для синхронізації програмного забезпечення між серверами програми використовуються вбудовані механізми IT-Enterprise. Серед усіх серверів додатків виділяється «головний сервер програми» (master-server), на якому завжди підтримується останній актуальний стан програмного забезпечення. Всі інші сервера додатків звіряють версію свого ПЗ з версією master-сервера, і виконують оновлення ПЗ при необхідності.

3.9 Використання робочої станції як локального сервера додатків

Один з варіантів установки сервера додатків-установка т. зв. "локального" сервера додатків. Локальний сервер додатків використовується тільки даною робочою станцією. У цьому випадку ПЗ на клієнтській станції включає в себе і сервер додатків і клієнтську частину.

Цей варіант рекомендується використовувати для розробників в тестовому комплекті, щоб їх локальні зміни не заважали іншим розробникам працювати з налагодженою системою. Варіант рекомендується використовувати тільки в поєднанні з варіантом «кілька серверів додатків». На рисунку 3.5 представлена можлива схема розподілу клієнтських робочих місць за серверів додатків. Два робочих місця в локальній обчислювальній мережі виступають як локальні сервера додатків. Інші робочі місця (в локальній і віддалених мережах) використовують виділені сервера додатків.

Переваги та недоліки: обчислювальна потужність робочої станції стає задіяною в роботі; зменшується навантаження на загальні сервери додатків;

працездатність або непрацездатність ПЗ на даній робочій станції ніяк не впливає на інші робочі станції і сервера додатків; робота в даному режимі можлива тільки в локальній обчислювальній мережі; для локального сервера додатків підходять не всі клієнтські операційні системи Windows.

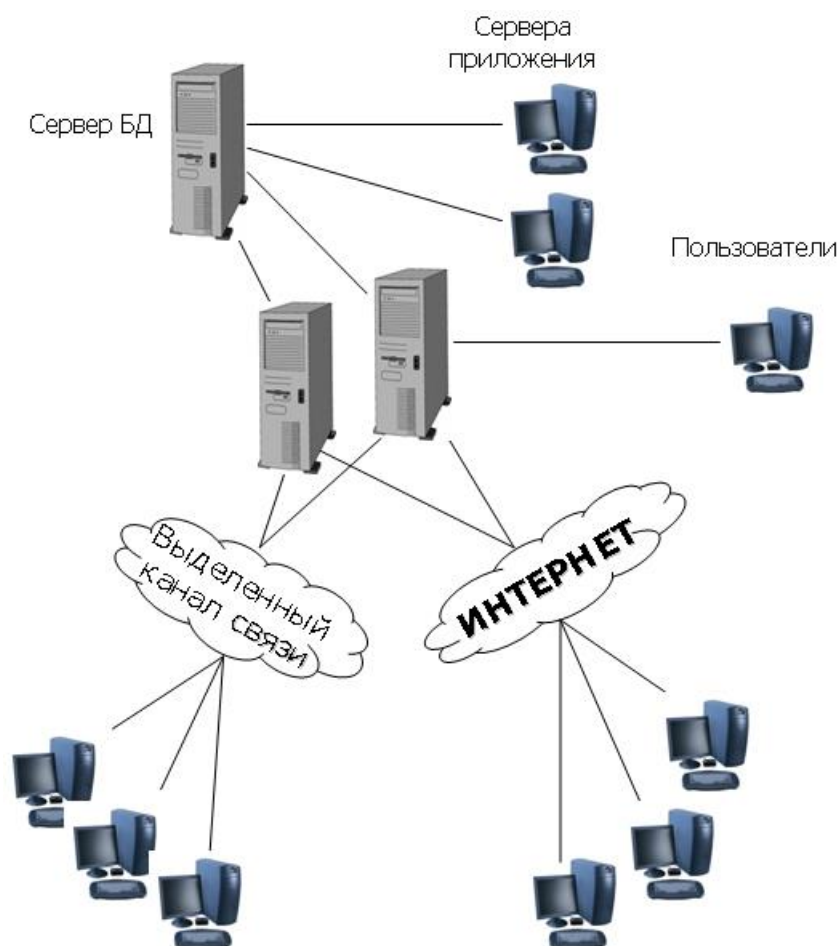


Рисунок 3.5 – Варіант «Робоча станція – сервер додатків»

3.10 Зв'язок компонентів системи

Всі компоненти спілкуються між собою за допомогою веб сервісів. Взаємодія з ERP комплектом відбувається за допомогою технології «Веб розрахунків», побудованих на основі технології ASMX, яка входить до платформи .NET. Існує і сучасніший фреймворк для побудови веб сервісів на платформі .NET - WCF, який має ряд переваг, таких як:

- різноманітні і гнучкі можливості транспорту;
- актуальність і розвиток технології;
- різноманітні варіанти хостингу.

Перевага WCF проявляється, наприклад, в тих випадках, коли вам потрібна висока швидкість транспорту, дуплекс, потокова передача, дотримання сучасних стандартів безпеки.

Але технологія ASMX все ще популярна, вона легка в застосуванні і прекрасно вирішує завдання в більшості випадків.

Переваги ASMX:

- простота розробки;
- проста у вивченні;
- відсутність проблем з конфігурацією (в порівнянні з WCF).

Обмін повідомленнями організований по протоколу SOAP (протокол обміну структурованими повідомленнями в розподілених обчислювальних системах, базується на форматі XML). Альтернативою даній технології є REST – архітектурний стиль взаємодії різних компонентів в мережі.

Основні відмінності SOAP і REST:

– SOAP активно використовує XML для кодування запитів і відповіді, а також сувору типізацію даних, що гарантують їх цілісність при передачі між клієнтом і сервером. З іншого боку, запити та відповіді в REST можуть передаватися в ASCII, XML, JSON або будь-якому іншому форматі та розпізнаються одночасно і клієнтом, і сервером. Крім того, в моделі REST відсутні вбудовані вимоги до типізації даних. В результаті пакети запитів і відповідей в REST мають набагато менші розміри, ніж відповідні їм пакети SOAP;

– У моделі SOAP рівень передачі даних протоколу HTTP є «пасивним спостерігачем», і його роль обмежується передачею запитів SOAP від клієнта серверу з використанням методу POST. Деталі сервісного запиту, такі як ім'я віддаленої процедури і вхідні аргументи, кодуються в тілі запиту. Архітектура REST, навпаки, розглядає рівень передачі даних HTTP як активного учасника

взаємодії, використовуючи існуючі методи HTTP, такі як GET, POST, PUT і DELETE, для позначення типу запиту до сервісу. Отже, з точки зору розробника, запити REST в загальному випадку більш прості для формулювання і розуміння, так як вони використовують існуючі і добре зрозумілі інтерфейси HTTP;

– Модель SOAP підтримує певний ступінь інтроспекції, дозволяючи розробникам сервісу описувати його API в файлі формату Web Service Description Language (WSDL, мова опису веб-сервісів). Створювати ці файли досить складно, проте це коштує витрачених зусиль, оскільки клієнти SOAP можуть автоматично отримувати з цих файлів докладну інформацію про імена і сигнатури методів, типів вхідний і вихідні даних. З іншого боку, модель REST уникає складнощів WSDL на догоду більш інтуїтивному інтерфейсу, заснованого на стандартних методах HTTP, описаних вище;

– В основі REST лежить концепція ресурсів, в той час як SOAP використовує інтерфейси, засновані на об'єктах і методах. Інтерфейс SOAP може містити практично необмежену кількість методів; інтерфейс REST, навпаки, обмежені чотири можливі операції, відповідні чотири методи HTTP.

Веб розрахунки - це функції, реалізовані всередині ERP системи стандартними засобами розробки системи (на мовах C #, Python, VFP), які беруть на вхід запити від зовнішніх додатків (параметри) і виконують будь-які функціональні дії в ERP системі, наприклад:

- перегляд поточних залишків на заданому складі;
- резервування товару;
- додавання заявки на закупівлю.

Перевагами веб розрахунків є:

– всі дії з даними виконуються всередині самої системи, а зовнішні програми не мають прямого доступу до даних;

– реалізуються стандартними засобами розробки системи ERP системи, їх конфігурація може виконувати відділ супроводу системи, без залучення сторонніх програмістів з підрядних організацій;

– публікація веб розрахунків - дозволяє викликати даний розрахунок сторонньому ПО, доступ до розрахунку (дозволений чи ні) задається адміністратором системи.

Сайт – клієнт обміну документами побудований основі технології ASP.NET MVC та Vue.js

ASP.NET – це технологія створення веб-застосунків і веб-сервісів від компанії Майкрософт. Вона є складовою частиною платформи Microsoft.NET, тому ця технологія найкраще підходить для нашого рішення, яке розробляється на мові С#, яка є складовою цієї ж платформи. ASP.NET має перевагу у швидкості в порівнянні з іншими технологіями, заснованими на скриптах [23].

Розширюваний набір елементів (рисунок 3.6) управління і бібліотек класів дозволяє швидко розробляти застосунки.

ASP.NET спирається на багатомовні можливості .NET, що дозволяє писати код сторінок на С#, VB, C/C++ та ін. Поділ візуальної частини та бізнес-логіки. Розширювана модель обробки запитів.

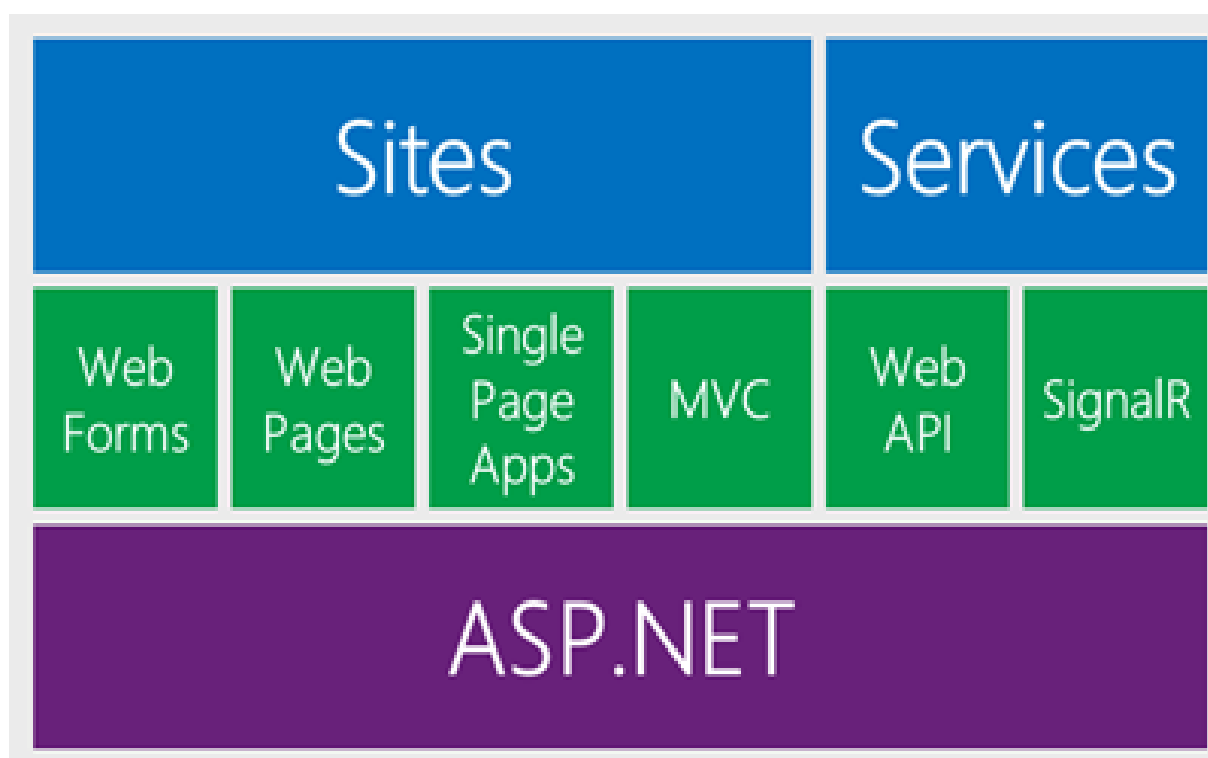


Рисунок 3.6 – Компоненти платформи ASP.NET

Vue.js — це прогресивний фреймворк для створення користувацьких інтерфейсів. На відміну від фреймворків-монолітів, Vue створений придатним для поступового впровадження. Його ядро в першу чергу вирішує завдання рівня подання (view), що спрощує інтеграцію з іншими бібліотеками та існуючими проектами. З іншого боку, Vue повністю підходить і для створення складних односторінкових додатків (SPA, Single-Page Applications), якщо використовувати його спільно з сучасними інструментами та додатковими бібліотеками.

Шаблон Архітектури Model-View-Controller (MVC) розділяє додаток на три основних компоненти: модель, представлення і контролер. ASP.NET MVC платформа являє собою альтернативу схемі веб-форм ASP.NET при створенні веб-додатків. ASP.NET MVC — легковагова платформа відображення з широкими можливостями тестування і, подібно до додатка на основі веб-форми, інтегрована з існуючими функціями ASP.NET, наприклад з головними сторінками і перевіркою справжності на основі членства.

MVC являє собою стандартний шаблон розробки, знайомий багатьом фахівцям. Деякі типи веб-додатків мають переваги при створенні на платформі MVC. Для інших може бути доцільно використання традиційної схеми додатків ASP.NET, заснованої на веб-формах і зворотній передачі. У деяких випадках можливе поєднання двох підходів: застосування однієї схеми не виключає використання іншого [24].

До складу платформи MVC входять наступні компоненти (рисунки 3.7):

- моделі — об'єктами моделей є частини програми, що реалізують логіку для домену даних програми (об'єкти моделей часто отримують і зберігають стан моделі в базі даних);
- представлення — служать для відображення призначеного для користувача інтерфейсу додатку (інтерфейс зазвичай створюється на основі даних моделі);
- контролери — здійснюють взаємодію з користувачем, роботу з моделлю, а також вибір представлення, що відображає призначений для користувача інтерфейс.

У додатку MVC представлення тільки відображає дані, а контролер обробляє дані, що вводяться і відповідає на дії користувача.

Наприклад, контролер може обробляти рядкові значення запиту і передавати їх в модель, яка може використовувати ці значення для відправки запиту в базу даних.

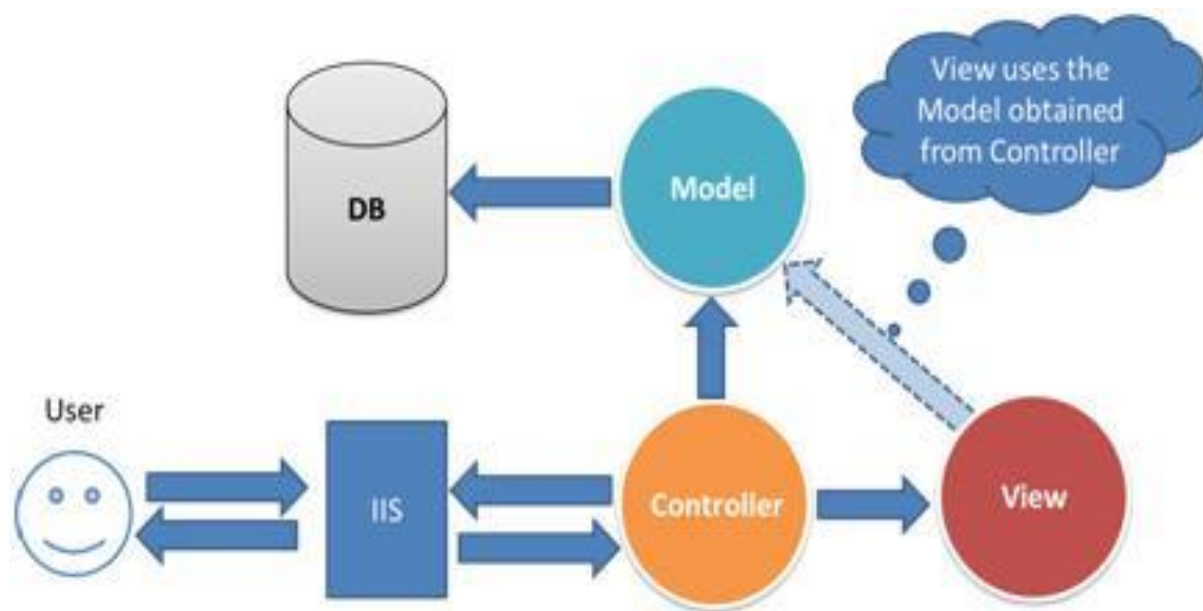


Рисунок 3.7 – Компоненти платформи ASP.NET MVC

Шаблон MVC дозволяє створювати додатки, різні аспекти яких (логіка введення, бізнес-логіка і логіка інтерфейсу) розділені, але досить тісно взаємодіють один з одним. Ця схема вказує розташування кожного виду логіки в додатку (рисунок 3.8). Інтерфейс розташовується в представленні. Логіка введення розташовується в контролері. Бізнес-логіка знаходиться в моделі. Цей поділ дозволяє працювати зі складними структурами при створенні програми, так як забезпечує одночасну реалізацію тільки одного аспекту. Наприклад, розробник може сконцентруватися на створенні представлення окремо від бізнес-логіки.

Зв'язок між основними компонентами програми MVC також полегшує паралельну розробку. Наприклад, один розробник може створювати представлення, інший - логіку контролера, а третій - бізнес-логіку моделі.

Платформа ASP.NET MVC має наступні переваги:

– полегшує управління складними структурами шляхом поділу додатку на модель, представлення і контролер;

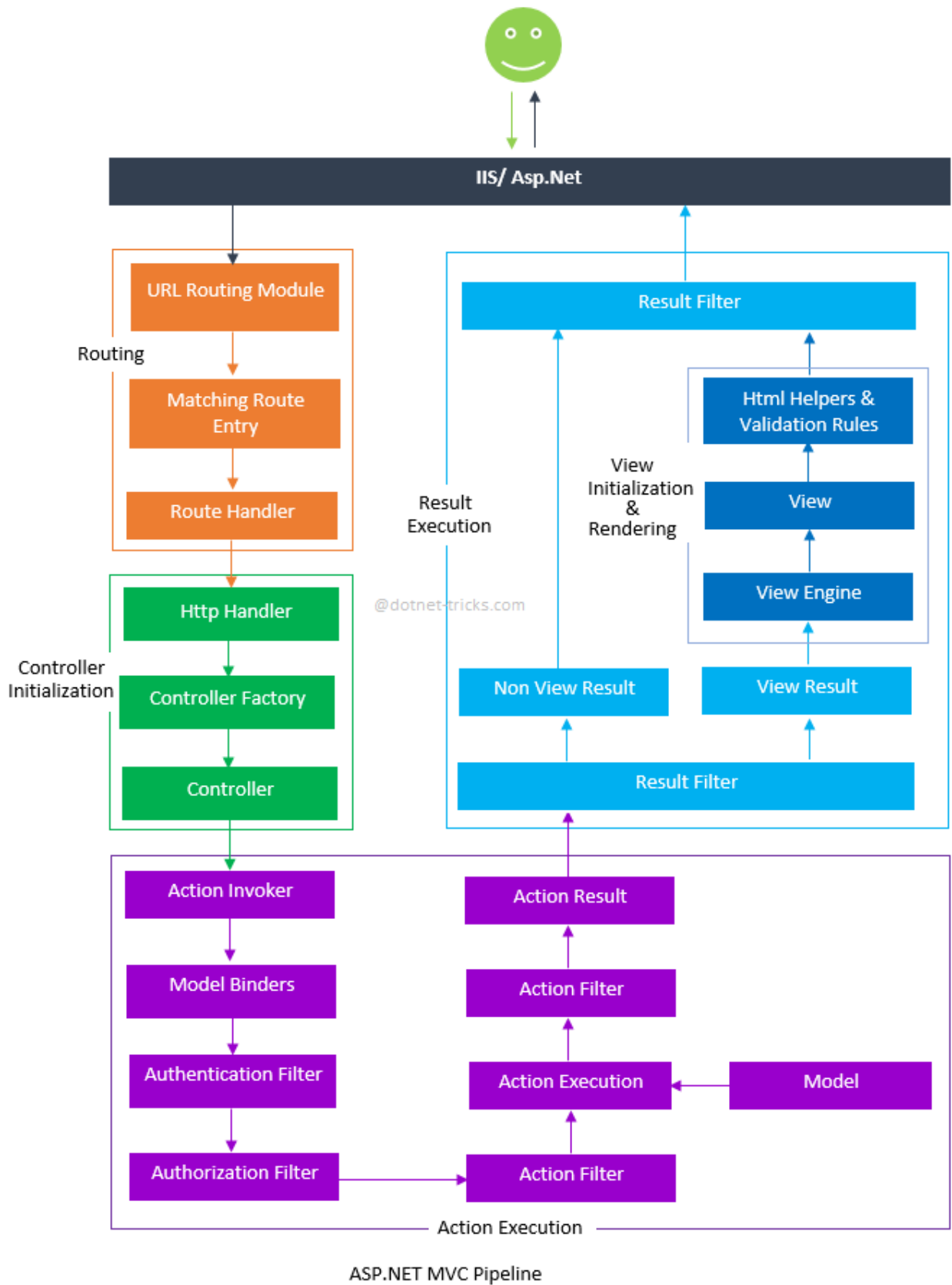


Рисунок 3.8 – Життєвий цикл сторінки ASP.NET MVC

- не використовує стан перегляду і серверні форми - робить платформу MVC ідеальною для розробників, яким необхідний повний контроль над поведінкою програми;
- використовує схему основного контролера, при якій запити веб-додатку обробляються через один контролер - дозволяє створювати додатки, що підтримують розширену інфраструктуру маршрутизації;
- забезпечує розширену підтримку розробки на основі тестування;
- добре підходить для веб-додатків, підтримуваних великими колективами розробників, а також веб-розробникам, яким необхідний високий рівень контролю над поведінкою програми.

3.11 Підтримка ЕЦП в web-додатках

В web-додатках підтримка ЕЦП реалізована двома шляхами: з використанням JavaScript-бібліотеки з використанням агента підпису. Користувач може вибирати реалізацію в залежності від типу ключа (файл або апаратний пристрій). Агент підпису підтримує як файлові ключі, так і апаратні пристрої. JavaScript-бібліотека через політик безпеки браузерів підтримує тільки файлові Ключі.

JavaScript-бібліотека виконана у вигляді JavaScript файлу, в якому реалізовані всі операції ЕЦП. JavaScript-бібліотека працює з використанням інтерпретатора JavaScript в наступних web-браузерах: Microsoft Internet Explorer 10 і вище, Mozilla Firefox 22 і вище, Google Chrome 28 і вище, Apple Safari 7 і вище.

Програмний комплекс агента підпису користувача ЦСК виконаний у вигляді окремого додатка, яке встановлюється в ОС, і JavaScript-обгортки.

Додаток працює під управлінням наступних ОС: Microsoft Windows XP / 2003 Server / Vista / 2008 Server / 7/8 / 8.1 / 2012 Server / 10 на PC (ПК) з 32-бітної або 64-бітної архітектурою; Linux / UNIX на PC (ПК) з 32-бітної або 64-бітної архітектурою, в т. ч. і під управлінням Sun OS на PC (ПК) з 64-бітної архітектурою SPARK; Apple MAC OS X 10.6 і вище на PC (ПК) з 64-бітної архітектурою.

Робота JavaScript-обгортки агента підпису підтримується в наступних web-браузерах: Microsoft Internet Explorer 10 і вище, Mozilla Firefox 22 і вище, Google Chrome 28 і вище, Opera 22 і вище, Safari 7 і вище.

Агент підпису користувача ЦСК – окремий виконуваний модуль, доступ до якого з web-браузера здійснюється з використанням технології JSON-RPC-запитів на основі протоколу HTTP(S) на локальному вузлі (IP-адреса - 127.0.0.1) безпосередньо із загальної JavaScript-обгортки.

4 ФОРМУВАННЯ МАТЕМАТИЧНОЇ МОДЕЛІ ТА ЗАГАЛЬНОЇ СТРУКТУРИ ПРОГРАМНОГО КОМПЛЕКСУ

Аутентифікація захищає двох учасників, які обмінюються повідомленнями, від впливу деякої третьої сторони. Однак проста аутентифікація не захищає учасників один від одного, тоді як і між ними теж можуть виникати певні форми суперечок [6]. У ситуації, коли обидві сторони не довіряють один одному, необхідно щось більше, ніж аутентифікація на основі загального секрету. Можливим рішенням подібної проблеми є використання цифрового підпису. Цифровий підпис повинний володіти наступними властивостями: повинна бути можливість перевірити автора, дату і час створення підпису; повинна бути можливість аутентифікувати вміст під час створення підпису; підпис повинен бути перевірений третьою стороною для вирішення спорів; таким чином, функція цифрового підпису включає функцію аутентифікації; на підставі цих властивостей можна сформулювати наступні вимоги до цифрового підпису; підпис повинен бути двійкового зразком, який залежить від підписується повідомлення; підпис повинен використовувати деяку унікальну інформацію відправника для запобігання підробки або відмови; створювати цифровий підпис має бути відносно легко; повинно бути обчислювально неможливо підробити цифровий підпис як створенням нового повідомлення для існуючої цифрового підпису, так і створенням помилкової цифрового підпису для деякого повідомлення.

У кожній країні існують власні закони щодо застосування та правомочності електронного підпису.

У загальному випадку електронні підписи можна розділити на 2 групи.

Група 1 – Загальновизнані електронні підписи. Підписи, правомочність яких визначена на законодавчому рівні. Документи, підписані такими електронними підписами, визнаються державними органами, судами і т. д. в тій же мірі, що і звичні паперові документи, підписані від руки. До реалізації підписів цієї групи висуваються жорсткі вимоги, виконання яких контролюється державними органами [7].

Група 2 – локальні електронні підписи. Підписи, для правомочності яких необхідно укладення договірних зобов'язань про правомочність підпису між суб'єктами договірних відносин. Такі підписи застосовуються для взаємовідносин між людьми, підприємствами, співробітниками і підприємством за обоюсторонньою домовленістю.

4.1 Опис основної моделі системи

Розглянемо, як відбувався обмін документами між підприємствами до автоматизації (рисунок 4.1):

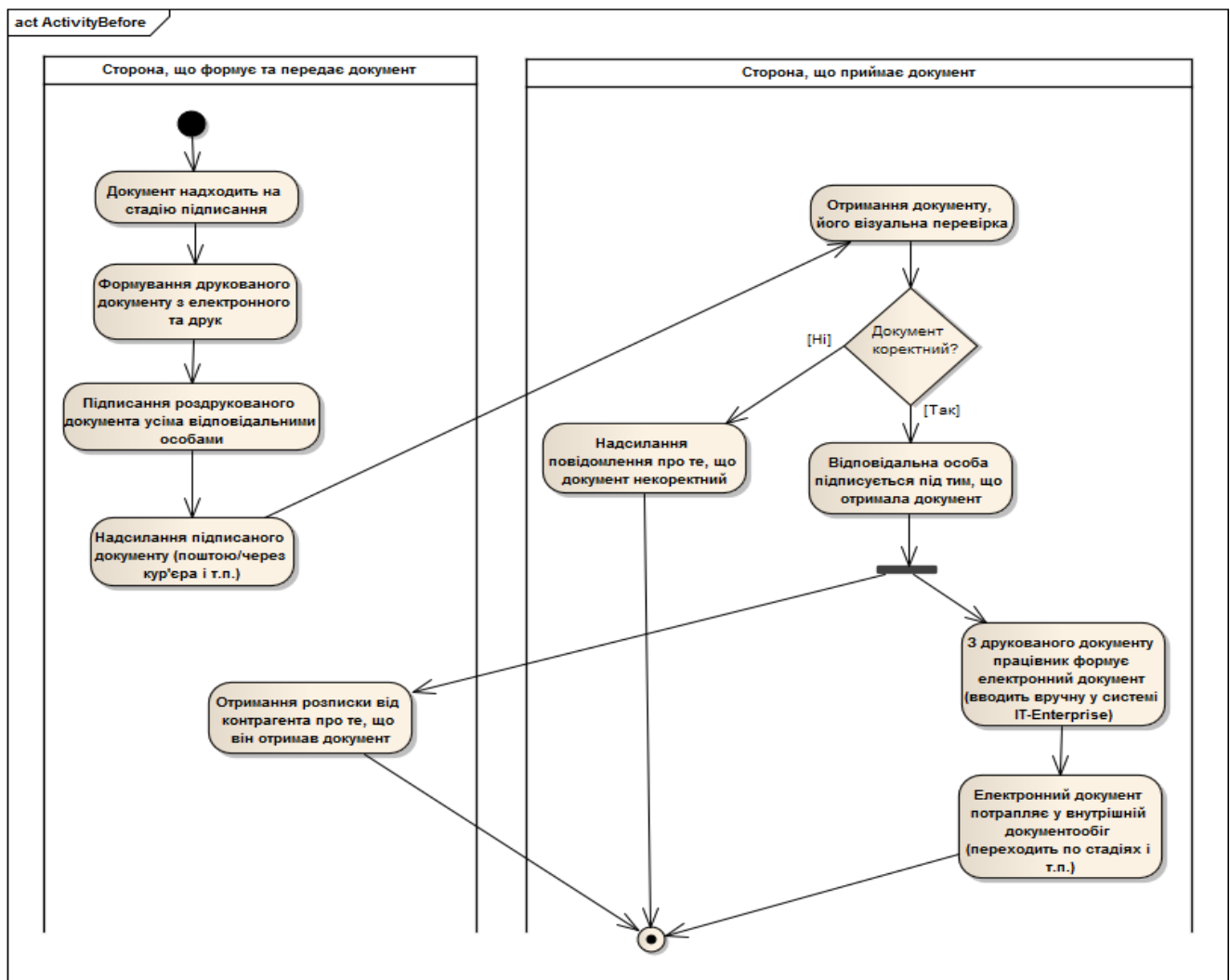


Рисунок 4.1 – Схема структурна діяльності процесу надсилання документа до автоматизації

Діаграма відображає процес формування документу, його надсилання та приймання.

Оскільки документ повинен мати юридичну силу – його друкують, і кожна із відповідальних осіб підписує його. Після цього документ надсилається тим чи іншим способом контрагенту. Варто звернути увагу, що після перевірки документу та підписання під його отриманням, контрагент вводить документ вручну у систему IT-Enterprise для того, щоб документ міг потрапити у внутрішню систему документообігу – передаватись по стадіях, затверджуватись і т.п.

Розглянемо тепер, як виглядатиме той самий процес після автоматизації (Рисунок 4.2):

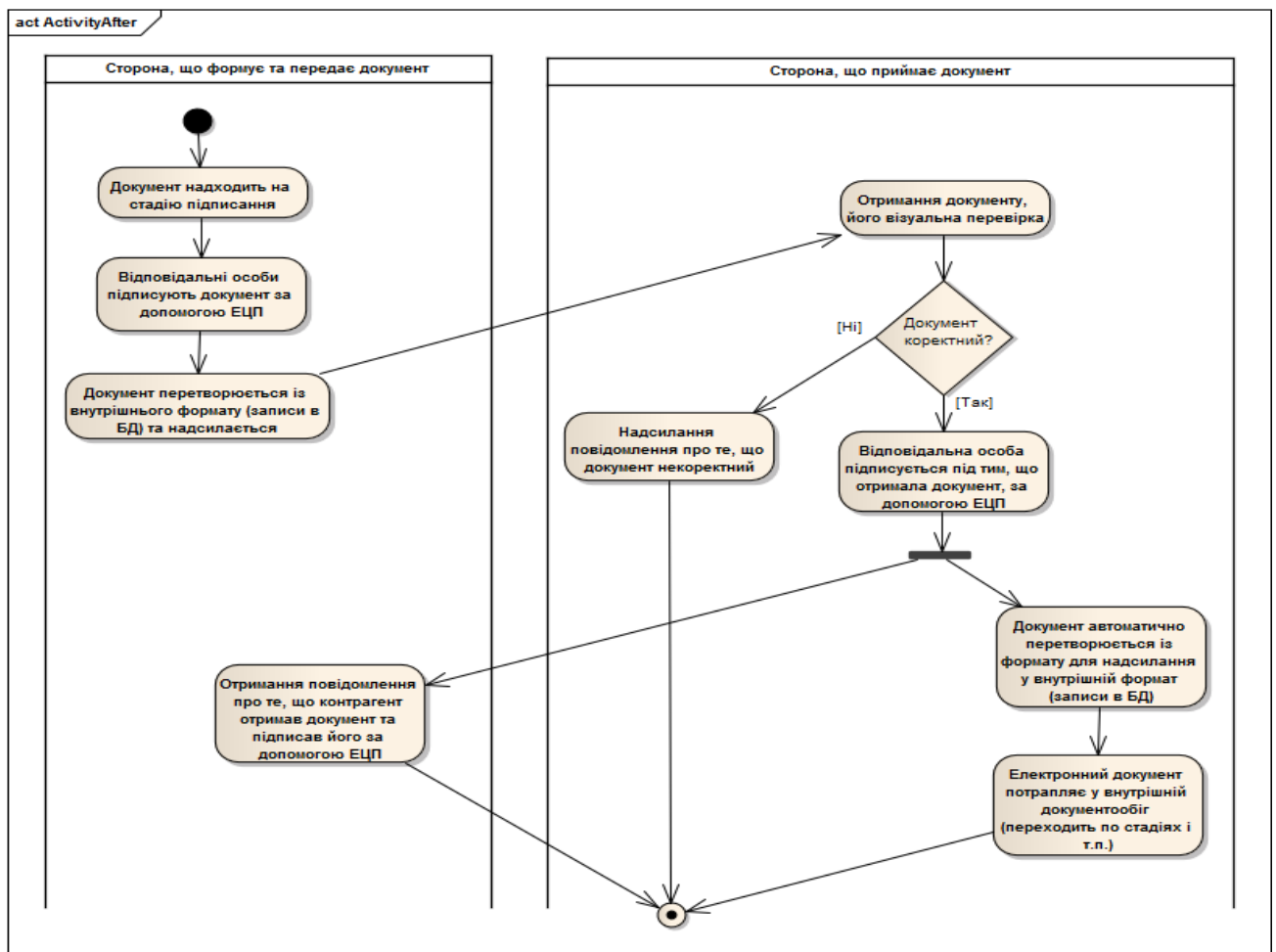


Рисунок 4.2 – Схема структурна діяльності процесу надсилання документу після автоматизації

Як бачимо, процес схожий, але є суттєві відмінності:

- усі підписи виконуються за допомогою юридично значимого ЕЦП. Це надає електронному документу, що надсилається, юридичної сили;
- відсутнє подвійне введення документу: на стороні, що надсилає, документ перетворюється в універсальний формат для надсилання, а на стороні, яка приймає документ, він перетворюється назад у внутрішній формат;
- усі процеси надсилання повідомлень автоматизовані (про те, що документ некоректний, про те, що документ отримано та підписано).

Суттєвим також є те, що усі ці процеси відбуватимуться у рамках однієї прозорої підсистеми, і надсилання та отримання документів легко буде контролювати.

4.2 Реалізація ЕЦП

В системі IT-Enterprise можуть застосовуватися електронні підписи як групи 1, так і групи 2.

Для роботи з електронними підписами як 1-ї, так і 2-ї групи використовується пара ключів, закритий і відкритий. Ці ключі генеруються програмою. Закритий ключ зберігається у користувача і нікому не передається, а відкритий реєструється в центрі сертифікації ключів.

Рішення про застосування підписів 1-ї або 2-ї групи слід приймати виходячи з розв'язуваної завдання бізнес.

На поточний момент робота з підписами групи 1 в системі IT-Enterprise реалізована для України. Згідно законодавства України для того, щоб електронний підпис мала юридичну силу, вона повинна відповідати в тому числі наступним вимогам: електронний цифровий підпис підтверджено з використанням посиленого сертифіката відкритого ключа за допомогою надійних засобів електронного підпису; під час перевірки повинен використовуватися посилений сертифікат ключа, чинний на момент накладення електронного цифрового підпису; секретний ключ передплатника повинен відповідати відкритому ключу, зазначеному в

сертифікаті; посилений сертифікат відкритого ключа повинен відповідати вимогам законодавства та бути виданий акредитованим центром сертифікації ключів; надійний засіб підтвердження електронного цифрового підпису, що має сертифікат відповідності або позитивний експертний висновок за результатами державної експертизи у сфері криптографічного захисту інформації [4].

ЕЦП, накладена в IT-Enterprise, може бути перевірена в будь-якому іншому інструменті. Так само, як і ЕЦП, накладена іншими інструментами, може бути перевірена в IT-Enterprise. При роботі з ключами 1-ї групи суті можливість використання апаратного рішення для генерації пари ключів і зберігання секретного ключа. Використання апаратного рішення дозволяє значно підвищити безпеку зберігання і виконання операцій з ключами – в цьому випадку секретний ключ ніколи не покидає фізичне пристрій, що виключає його крадіжку, чи дублювання.

Пристрій для реалізації апаратного рішення виконується у вигляді електронного USB ключа, зовні схожого на флешку. Це пристрій має внутрішні обчислювальні потужності і програмне забезпечення. Всі криптографічні операції виконуються ізольовано всередині пристрою, що значно підвищує безпеку рішення в цілому.

Надійність ЕЦП строго математично доведена. Питання надійності ЕЦП, вірності математичних доказів і т. п. не входять в предмет обговорення даного документа. Існують науково-дослідні інститути, які доводять надійність застосовуваних алгоритмів. Тільки алгоритми, що пройшли експертизу і численні перевірки, допускаються до застосування в ЕЦП як державними, так і міжнародними органами стандартизації.

Існує безліч алгоритмів для ЕЦП. У державних органах допускається використовувати тільки алгоритми згідно з державними стандартами (причому, алгоритми в різних країнах різні).

Для IT-Enterprise застосовуються загальноприйняті стандартні алгоритми асиметричного шифрування, реалізовані компанією Microsoft в криптопровайдері Windows.

4.3 Реалізація модуля управління файлами

Даний модуль дозволяє прив'язувати до певного документу системи (договір, наказ, особиста картка, технологічний процес тощо) довільні електронні файли (текстові документи, картинки, фотографії та інші). Ці файли в системі носять назву «Вкладення». Це дозволяє в потрібний момент звернутися за відсутньою інформацією оригінальним договором або наказом, переглянути специфікацію цікавить виробу або будь-яку іншу нормативну інформацію по даному документу без звернення до яким-небудь додатковим довідників або картотек.

Як вкладення може виступати комп'ютерний файл будь-якого формату. Для структуризації вкладень в межах документа використовуються види вкладень. Види вкладень також визначають формат файлів, можливість використання шаблонів. Система забезпечує поділ прав доступу до вкладень.

Можливість роботи з вкладеннями доступна в будь-якій функції системи. Для кожного типу документів системи виділяють свій тип вкладень. Тобто документів певного типу відповідають тільки вкладення даного типу.

В системі IT-Enterprise підтримується робота з вкладеннями двох типів: прості вкладення і вкладення з заданими видами.

Прості вкладення – інструмент для прикріплення до будь-рядок будь-якої таблиці будь-якого заданого користувачем файлу. Даний інструмент не вимагає ніякої додаткової настройки. Можливість працювати з простими вкладеннями з'являється в панелі інструментів автоматично. Даний механізм дозволяє кожному користувачеві прикріпити файли до будь-рядок будь-якої таблиці, переглядати будь-які вкладення, прикріплені іншими користувачами, видаляти власні вкладення. Копії файлів можуть зберігатися в базі даних або на мережевому ресурсі.

Вкладення з заданими видами – це розширений механізм роботи з вкладеннями. Для таких вкладень існує можливість налаштування доступу на перегляд, додавання, видалення і коригування. Також для вкладення із заданим видом можна задати пароль на перегляд документа. Даний механізм передбачає

можливість зберігання вкладень в базі даних або на файловому ресурсі або як посилання на вже існуючі файли.

4.4 Конструктор управления контентом

Для налаштування структурованого зберігання вкладень використовується функція (рисунок 4.3) «Конструктор управління контентом» (доступна в меню Адміністрування системи → Керування контентом → Керування контентом → Конструктор управління контентом).

Вид	Наименование	Тип файла	Хранить
DLRKSD	Документы в формате DOC	DOC	в базе данных
DOC_1	Шаблон исходящего документа	DOC	в базе данных
JPG	Электронная копия документа	JPG	в базе данных
PDF	Договора	PDF	в базе данных

О	N	Вид	Статус	Ключ	Доступ для	Группы	Пользователь	Просм	Корр	Доб	Удал
13		DLRKSD			Всех пользователей			+	+	+	+
15		JPG			Всех пользователей			+	+	+	+
16		PDF			Всех пользователей			+	+	+	+
18		DOC_1			Всех пользователей						

Рисунок 4.3 – Конструктор управління контентом

4.4.1 Прив'язки

Для можливості підключати вкладення до рядків таблиці цю таблицю необхідно описати в таблиці прив'язок. Прив'язки вкладень для системних таблиць налаштовуються розробником і не можуть змінюватися користувачами на об'єкті. Користувачі на об'єкті можуть додавати свої таблиці прив'язок для прив'язки

вкладень до рядків користувальницьких таблиць. При описі таблиці прив'язок вказуються наступні параметри (рисунок 4.4).

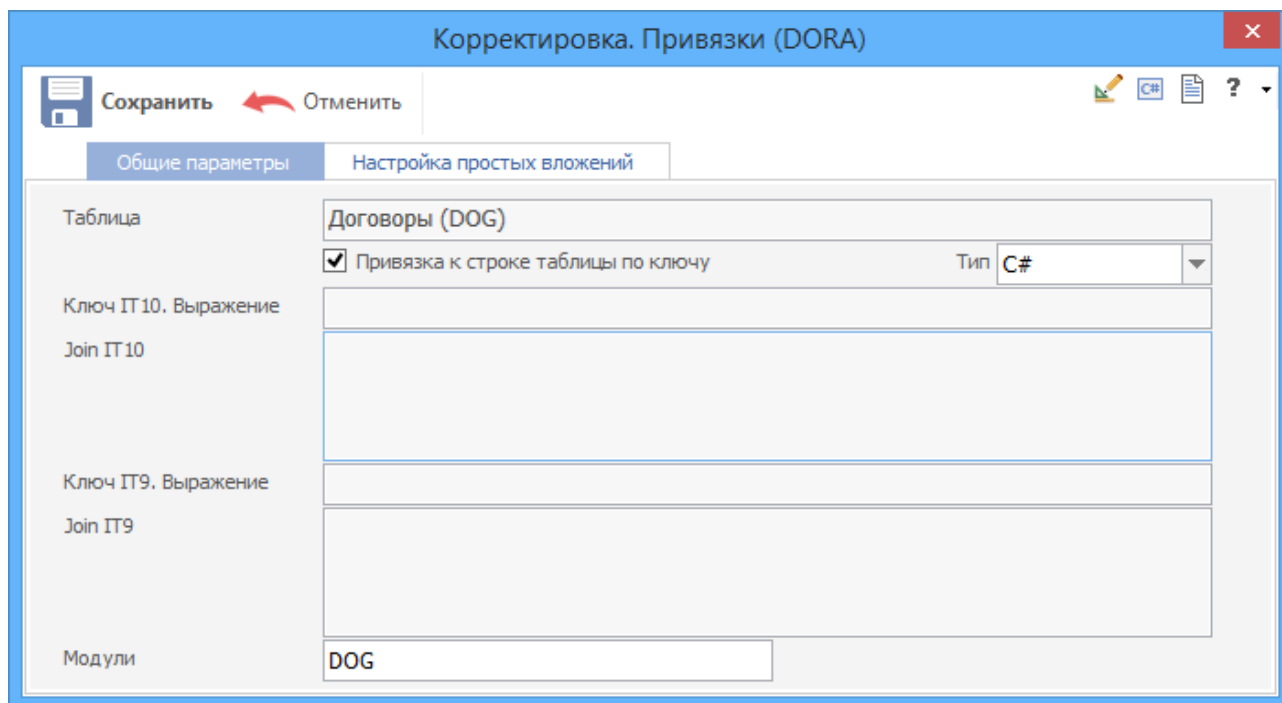


Рисунок 4.1 – Екран для опису прив'язки вкладень. Закладка «Загальні параметри»

Таблица – таблиця прив'язки вкладень. Прив'язка до рядка таблиці по ключу – ознака прив'язки вкладень до рядка таблиці за ключовим виразом. Якщо ознака встановлений для типу «C#», то вказувати вираз ключа не потрібно, воно буде визначено автоматично по опису ключових полів таблиці. Тип VFP/VFP+C#/C# – тип прив'язки вкладень. Використовується 3 типи: VFP - для прив'язки вкладень у веденнях версії 9; VFP + C # - для прив'язки вкладень, які використовуються у веденнях і версії 9 і версії 10; C # - для прив'язки вкладень у веденнях версії 10. Вирази ключа IT10-вказується вираз ключа для прив'язки вкладень на мові Python. Використовується для типу VFP+C# або C#. Приклад: `sqlApi.Str ("IBM.IBM_SYS", 8)`. Join IT10 – вказується вираз join на мові Python. Вираз може використовуватися для визначення складного ключа, який будується за кількома таблицями.

Використовується для типу VFP+C# або C#. Приклад: "JOIN DMT on DMZ.KDMT=DMT.KDMT". Вирази ключа IT9 – вказується вираз на мові Foxpro. Приклад: Str(UnDoc,10). Модулі – перелік модулів, в яких використовується таблиця прив'язок вкладень.

Також при описі прив'язки вкладень можна налаштувати правила для використання простих вкладень. Для цього використовується закладка "Налаштування простих вкладень» (рисунок 4.5).

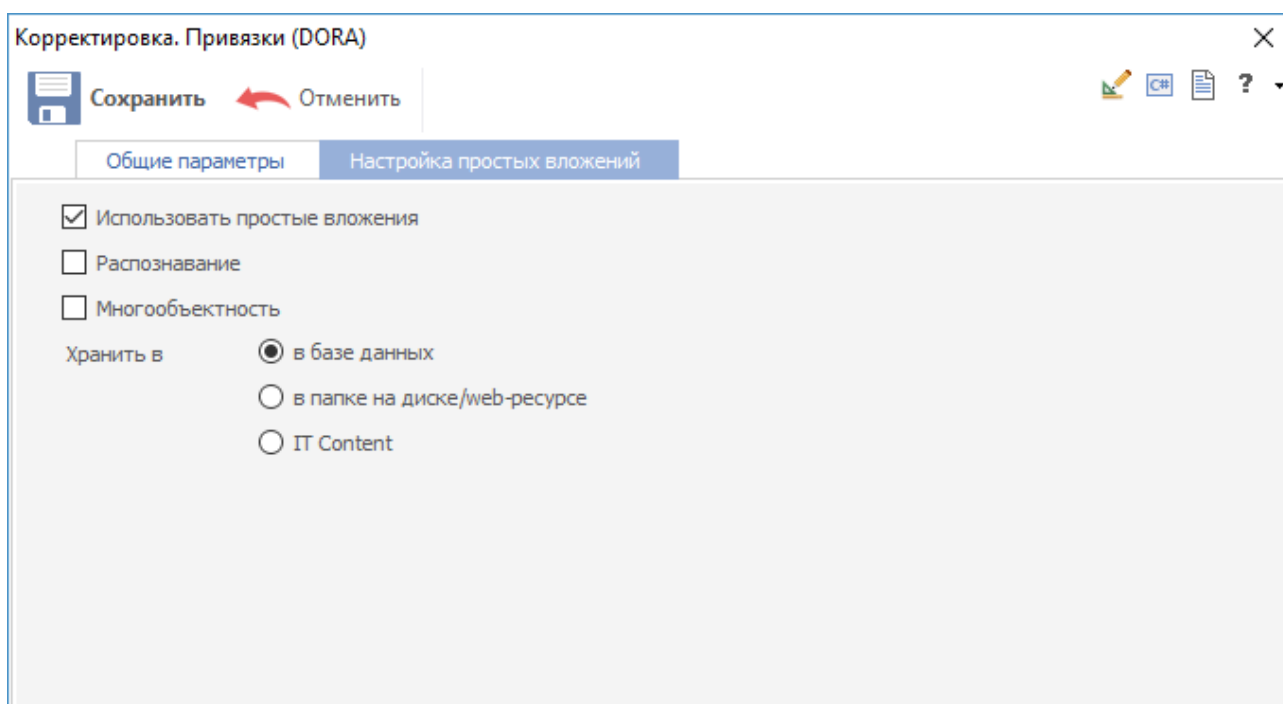


Рисунок 4.2 – Экран для опису прив'язки вкладень. Закладка "Налаштування простих вкладень»

Використовувати прості вкладення – ознака використання простих вкладень для заданої таблиці прив'язок. Якщо для даної таблиці прив'язок необхідно використовувати тільки вкладення з певними видами (з налаштованим доступом, типами файлів тощо), то за допомогою даної настройки можна відключити використання простих вкладень.

Розпізнавання – виконувати розпізнавання файлів після додавання (детально в розділі «Розпізнавання тексту»). Багатооб'єктність - при додаванні вкладення

буде зафіксовано об'єкт обліку, під яким працює користувач. Подальша робота з вкладенням буде доступна тільки в цьому об'єкті обліку. Зберігати в – визначає місце зберігання простих вкладень. За замовчуванням всі прості вкладення зберігаються в базі даних. За допомогою даної настройки можна задати каталог на мережевому ресурсі, де будуть зберігатися вкладення (рисунок 4.6).

Корректировка. Привязки (DORA)

Сохранить Отменить

Общие параметры IT9

Наименование: Делопроизводство. Документы

Длина ключа: 10

Наименование ключа: 'N '+allt(NDM)+' от '+DTC(DDM)+' о '+allt(TEXT_S)

Обратный поиск:

☐ Вложения верхнего уровня доступны ниже по своему ключу

☐ Вложения нижнего уровня доступны выше по своему ключу

Перечень полей: [DLRKS.NDM, DLRKS.DDM, DLRKS.TEXT_S]

Выражение для Browse: [NDM :h="Номер" :15,
DDM :h="Дата" :11,

Выражение для AftSelect:

Рисунок 4.3 – Экран для опису прив'язки вкладень. Закладка «IT9»

Закладка «IT9» доступна тільки для прив'язок вкладень типу «VFP» або «VFP+C#». Всі налаштування на цій закладці використовуються тільки для вкладень, які використовуються у веденнях версії 9.

4.4.2 Види вкладень

В описі виду вкладення вказується місце зберігання вкладення, формат файлів вкладень, шаблон (рисунок 4.7).

Корректировка. Виды (DORV)

Сохранить Отменить

Основные параметры IT9 IT10

Вид: DMZ ☐ Системная строка

Наименование: Документы движения ТМЦ

Раздел: Материальные ценности

Тип файла: BMP

Вложения

Хранить: ☒ в базе данных ☐ в папке на диске/web-ресурсе ☐ как ссылку ☐ IT Content

Шаблон

Загрузить Выгрузить Просмотр Изменить Удалить

☐ Распознавание

☐ Многообъектность

Условие

Сортировка

После добавления: Ничего не делать

После корректировки: Сохранить

Описание

Рисунок 4.4 – Экран коригування виду вкладення. Закладка «Основні параметри»

Для каждого вида вкладення необхідно вказати наступні показники: вид – унікальний код виду вкладення, найменування – найменування виду вкладення та версія виду вкладення – ознака використовується для можливості перегляду вкладень за шаблонами з виразами на мові VFP в інтерфейсах С#: для всіх видів вкладень v9, при описі яких використовувався VFP-код, необхідно в ознаку "версія" Встановити рівним " VFP", якщо вкладення не використовувався vfp-код, ознаку "Версія" можна залишити порожнім – вкладення буде доступно для роботи

як на платформі v9, так і на платформі v10. Розділ – розділ вкладень згідно з довідником «Розділи документів». Тип файла – розширення файлу вкладення, що описує його формат для програм перегляду. Приклади типів файлів: TXT, DOC, RTF, BMP, JPG, CDR і т. д. Режим зберігання вкладення – можливі наступні режими зберігання: в базі даних – вкладення зберігається в полі бази даних; у папці на диску/хмарі – вкладення зберігається в окремому файлі на диску або на інтернет-ресурсі (порталі SharePoint або Ftp-сервері). Якщо даний файл будуть переміщений або видалений, то неможливо буде переглянути даний файл з системи; як посилання – використовується для можливості зберігання у вкладеннях посилань на існуючі файли.

Особливості роботи: при додаванні вкладення після вибору файлу буде додано посилання на цей файл; для вкладення-посилання доступні стандартні можливості перегляду / коригування; при видаленні вкладення-посилання видаляється опис посилання, оригінальний файл на мережевому ресурсі не видаляється.

IT content – зберігання вкладень в окремій базі даних, призначеної для зберігання файлів. Ця база використовується для оптимізації використання місця в основній базі даних IT-Enterprise.

Шаблон – файл, який буде підставлятися в якості вкладення «за замовчуванням». Тобто якщо не заданий який-небудь інший файл. Доступні наступні кнопки роботи з шаблоном вкладення:

- завантажити – завантажити шаблон вкладення в поле бази даних;
- вивантажити – зберегти шаблон вкладення на диск;
- перегляд – перегляд файлу шаблону вкладення;
- видалити – видалити шаблон вкладення.

Умова – умова, при якому можна використовувати даний вид вкладення для рядка документа системи. В умови доступні значення полів оригінальної таблиці, для якої прикріплюється вкладення. Використовується тільки для вкладень в інтерфейсах версії 10.

Дія після додавання - можна вибрати дію, яку необхідно виконати з вкладенням після додавання. Наступні дії доступні: нічого не робити – після додавання вкладення не виконувати ніяких дій (за замовчуванням); викликати для коригування - після додавання вкладення файл вкладення буде відкритий додатком за замовчуванням для можливості коригування; дія після коригування - можна вибрати дію, яку необхідно виконати з вкладенням після коригування. Наступні дії доступні: зберегтиС зберегти внесені зміни (за замовчуванням) та створити нову версія – після коригування буде додана нова версія.

Перегляд і коректування вкладень, які використовуються у версії 10, здійснюється за допомогою програми за промовчанням у налаштуваннях операційної системи для даного типу файла.

Для виду вкладень, які використовуються у веденнях версії 9, можна задати наступні налаштування (рисунок 4.8).

Шаблон для вкладень версії 9 можна зберігати або в базі даних або в папці на диску. Шаблон для вкладень версії 10 зберігається тільки в базі даних. Переглядач – аналогічно редактору вказується ім'я параметра локальної установки, в якій зберігатися програма редагування файлів даного типу для локального комп'ютера. Аналогічно відбувається пошук переглядача при порожньому значенні. Так само допускається використання функції користувача. Редактор – вказується ім'я параметра локальної установки, в якій зберігатися програма редагування файлів даного типу на локальному комп'ютері, наприклад: CFGEdiWTXT, CFGEdiWDOC. Якщо явно не вказано параметр, то при відповідному виклику з вкладень буде підставлений локальний параметр відповідний зазначеному вище типу файлу. При необхідності, в якості редактора можна використовувати функцію користувача, наприклад, в «Кадрах» часто використовується функція =WordDocum() для роботи з макропідстановками в документах Word. При вказівці функції першим символом обов'язково має бути рівність('='). В якості редактора можна вказувати тимчасову локальну змінну. Також при завданні редактора можна використовувати макропідстановку (наприклад, &CfgEdiWDir+'\\Notepad').

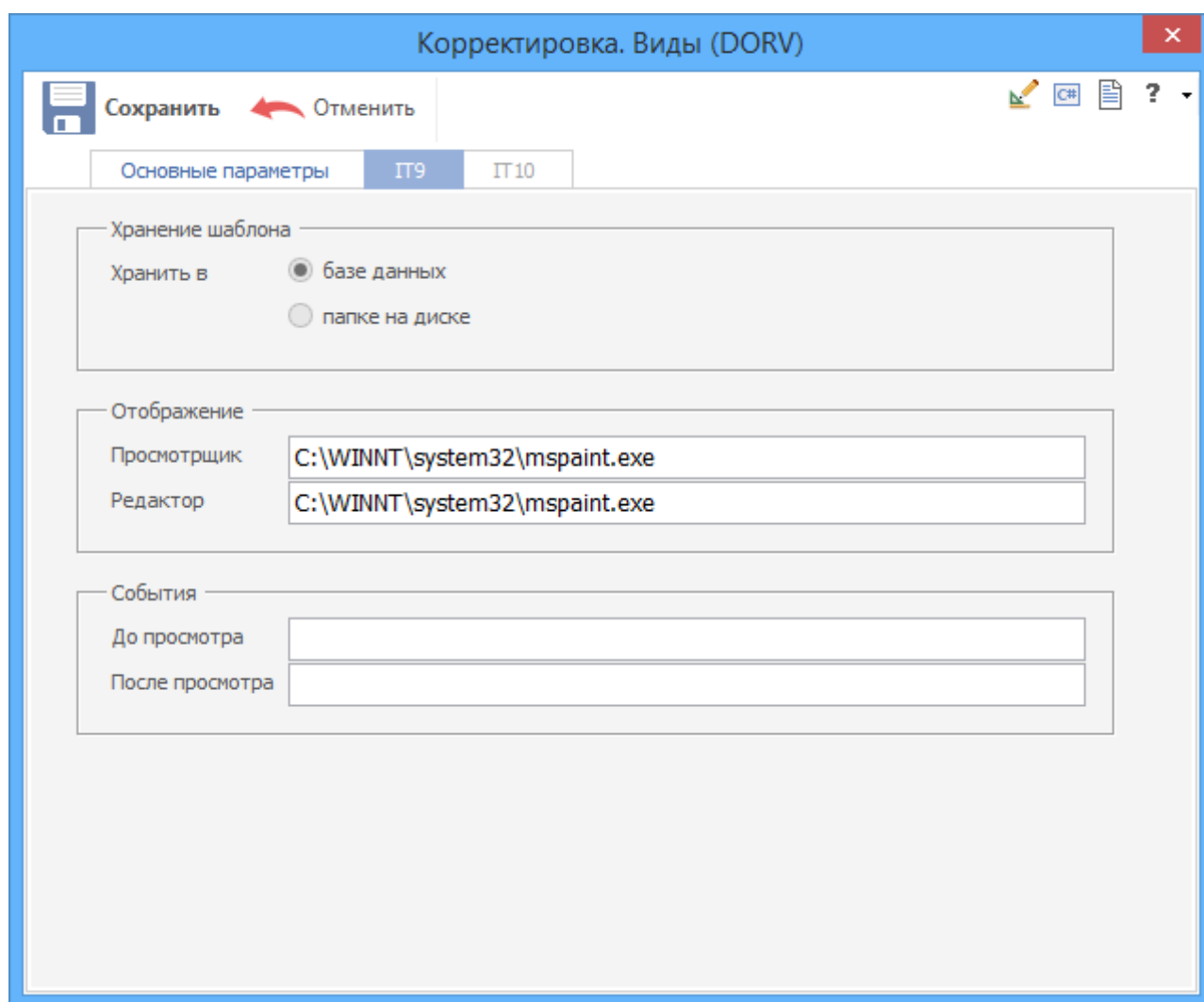


Рисунок 4.5 – Экран коригування виду вкладення. Закладка «ІТ9»

До перегляду – вказується процедура користувача, яка викликається перед переглядом/коригуванням вкладення. Для визначення режиму роботи перегляд/коригування необхідно аналізувати змінну `m.choice`. При перегляді `m.choice = ASC('A')`, при коригуванні `m.choice = ASC('B')`. Ім'я файлу вкладення, яке потім виводиться на екран, зберігається в змінній `m.FilePict`. Після перегляду – вказується процедура користувача, яка викликається після перегляду / коригування вкладення. При роботі з мемо-полями, файл на який посилається `m.FilePict` після перегляду/коригування видаляється.

Для вкладень, які використовуються у веденнях версії 9, є можливість налаштовувати різні редактори для перегляду і коригування. Для кожного виду вкладень є можливість вказати свій редактор на перегляд і коректування. Ім'я або

шлях до редактора вказується у вікні коригування видів вкладень у відповідному розділі. В поле редактор-вказується програма, що викликається при редагуванні вкладення (F11), в поле переглядач-вказується програма, що викликається при перегляді вкладення (F10). В якості редактора/переглядача можна вказувати безпосередньо шлях до виконуваного файлу прикладної програми. Наприклад, "D:\Program Files\ACD Systems\ACDSee\ACDSee.exe". У даному випадку, слід врахувати, що зазначена програма повинна розташовуватися на одному і тому ж диску і в одному і тому ж каталозі на всіх комп'ютерах або, як шляху, повинен бути вказаний мережевий ресурс однаково доступний для всіх локальних комп'ютерів.

Параметр локальної установки – системний, або призначений для користувача. Наприклад, CFGEdiWBMP. В даному випадку програма перегляду/редагування буде братися з зазначеного локального параметра. Т. к. для кожного робочого місця локальні параметри можуть відрізнятися, то при налаштуванні через локальні параметри, на різних комп'ютерах можуть бути вказані різні шляхи до програм перегляду або взагалі різні програми перегляду. Даний варіант краще першого, тому що не прив'язаний до конкретного шляху, або мережевого ресурсу. Але вимагає коректної настройки локальних параметрів на всіх комп'ютерах.

Якщо в локальному параметрі вказано порожнє значення. Є можливість не вказувати в локальному параметрі конкретний редактор. В даному випадку буде визначена і запущена програма, яка в операційній системі «за замовчуванням» відкриває файли даного типу (з даним розширенням) на перегляд/коригування.

Порожнє значення – в даному випадку по розширенню файлу визначається відповідний локальний параметр. Наприклад, розширення файлу .bmp, відповідає локальний параметр CFGEdiWBMP. Якщо відповідний локальний параметр не знайдений, то визначається редактор по налаштуваннях операційної системи. Даний режим аналогічний попередньому, але більш підходить для видів вкладень з нефіксованим (довільним) розширенням файлів. Тоді програма перегляду визначається динамічно, в залежності від прикріпленого файлу.

Процедура перегляду/коригування – будь-яка процедура стандартна, або призначена для користувача, написана на Visual FoxPro. При вказівці процедури першим символом обов'язково має бути рівність('='), наприклад, =WordDocum(). З системою поставляється процедура =WordDocum(). Яка запускає Microsoft Word як OLE-об'єкт і виконує макропідстановку даних з робочої області .doc-файл.

Використання макропідстановки – при описі редактора можна використовувати макропідстановку, наприклад, &CfgEdiWDir+'\\Notepad'. У макропідстановці можна використовувати будь-які змінні, визначені в системі. Підстановка обов'язково повинна починатися на &.

Види вкладень з довільним розширенням файлів – є можливість не фіксувати стандартне розширення для конкретного виду вкладень. Для цього у вікні редагування виду вкладень поле "тип файлу" не заповнюється. Тоді, в якості вкладень даного виду можна підключати в систему будь-які файли. Для вкладень даного виду рекомендується не заповнювати редактор і переглядач. Тоді відповідні програми перегляду будуть визначатися по розширенню прикріпленого файлу.

4.4.3 Статуси видів вкладень

Статуси призначені для обліку поточного стану вкладення. Статус вкладення є необов'язковою характеристикою і може вводиться вибірково для певних видів документів. Основне призначення статусу – можливість змінити права доступу користувачів до певного вкладення відразу для групи користувачів. Тобто ми маємо можливість, заздалегідь налаштувати права користувачів для різних статусів (наприклад, "проект" і "затверджений"), а потім просто змінювати статус електронного вкладення документа.

Для кожного виду зазначаються свої власні статуси, причому, якщо для різних видів збігаються коди статусів, то при зміні виду вже існуючого вкладення, його статус не змінюється.

При описі статусу для виду вкладення вказується наступна інформація (рисунок 4.9):

Таблица	Вид вложения	Статус	Наименование	Описание
DMZ	Документы движения ТМЦ (DMZ)	OKD	Утверждено	

Рисунок 4.6 – Экран коригування статусу вкладки

- вид вкладки – вид, для якого описаний статус;
- вид вкладки – вид, для якого описаний статус;
- статус – унікальний код статусу вкладки;
- найменування – повне найменування статусу;
- опис – опис статусу.

4.4.4 Налаштування доступу до вкладень

Перед роботою з вкладеннями документів необхідно налаштувати права користувачів. Налаштування доступу здійснюється тільки для вкладень із заданими видами. Існує можливість налаштування спільного доступу до всіх вкладень певного виду, або відразу до всіх вкладень одного типу. Також є можливість налаштування доступу до вкладень певного документа системи або поділ прав доступу за видами вкладень в межах одного документа.

Налаштування прав доступу до вкладень має прийнятий в середовищі Windows вид і працює за принципом пріоритету "заборони" над "дозволом". Тобто якщо користувачу був заборонений доступ до даного вкладки за цим документом, то, незважаючи на можливе існування записів дозволяють роботу з

даним вкладенням, поки не буде знято заборону, користувачеві буде закритий доступ до даного вкладення.

Для кожного режиму роботи з вкладеннями: перегляд, коригування, додавання, видалення фактично можна вказати три різних варіанти прав доступу.

Заборонений – якщо вказано цей прапор, користувачеві буде завжди заборонено дану дію, на тому діапазоні вкладень, який визначається рядком.

Дозволений – якщо вказано даний прапор, користувачеві буде дозволено дану дію на всьому зазначеному діапазоні вкладень, за винятком тих вкладень, для яких дана дія явно "заборонено" іншими рядками настройки доступу.

Не визначено доступ (не вказано ні той, ні інший прапор) – для вкладень, що охоплюються поточним рядком, дана дія як не дозволяється, так і не забороняється. Тобто доступ буде визначатися іншими рядками. Якщо іншими рядками не буде дозволений доступ до даного вкладення, то дане вкладення буде недоступне.

При коригуванні / додаванні нового запису (додавання нових прав) вказується наступна інформація (Рис. Вісім).

Налаштування доступу-область дії даних прав:

- вкладення по таблиці;
- вкладення по таблиці заданого виду – права для зазначеного виду вкладень;
- вкладення по таблиці заданого виду і статусу-права;
- прості вкладення.

Вид вкладення – для налаштування на вказаний вид або статус. Статус вкладення – для налаштування на вказаний статус вкладення; період дії – період дії налаштованих прав доступу; ознака дії прав доступу; права доступу-дозволи або заборони для режимів перегляду, коригування, додавання, видалення і налаштування прав до вкладення; пріоритет дозволу за замовчуванням заборона має пріоритет над дозволом (якщо є хоч один заборона для дії, то доступ до дії буде заборонений), налаштування використовується для встановлення пріоритету дозволу над заборонами (рисунок 4.10).

Корректировка. Вложения. Доступ (DORD)

Сохранить Отменить

Настройка доступа Вложения по таблице

Период действия 16.10.2014 -

Права

	Разрешено	Запрещено
Просмотр	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Корректировка	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Добавление	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Удаление	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Настройка прав	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Приоритет разрешения	<input type="checkbox"/>	<input type="checkbox"/>

☒ Все
☐ Группа
☐ Пользователь

Примечание

Рисунок 4.7 – Налаштування прав доступу

Для кого встановлюються дані права:

- користувач – права діють для окремого користувача системи;
- групи користувачів – перелік груп користувачів;
- всі – для всіх користувачів системи.

4.4.5 Шаблони вкладень

Для вкладень із заданими видами існує можливість налаштувати шаблон, який буде використовуватися при додаванні вкладення. Шаблон-це документ із заданою структурою, розділами, абзацами та форматуванням. В якості шаблону використовуються документи з розширенням .docx (для вкладень з прив'язками типу VFP з розширенням .doc).

При створенні шаблону необхідно створити docx-документ заданої структури з макроподстановками. Як макропідстановки використовуються вирази на мові Python (для вкладень з прив'язками типу VFP на мові Visual FoxPro). За допомогою

макропідстановок задаються частини документа, які змінюються для різних документів (дата документа, відповідальний та інші реквізити).

Для виразів в макропідстановках можна використовувати всі поля таблиці прив'язки вкладень через префікс " m.» (m.ім'я_поля). Також існує можливість визначити додаткові змінні. Для цього потрібно в бізнес-логіці виду вкладень перевизначити метод GetAdditionalValues(). У цьому методі необхідно заповнити об'єкт з полями <змінна><значення>. Створені таким чином змінні потім можна використовувати в виразах в тексті шаблону з префіксом " m»".

На рисунку 4.11 представлена частина налаштованого шаблону.

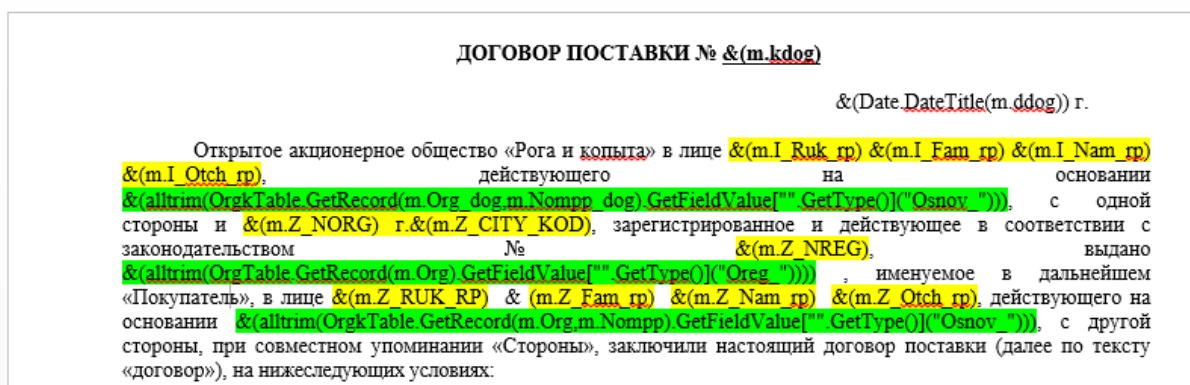


Рисунок 4.8 – Налаштування шаблону для виду вкладень

Після створення шаблону документа (.docx) необхідно завантажити його в опис виду вкладення. В описі виду вкладення є можливість переглядати, коригувати або видалити доданий шаблон.

При додаванні вкладення з заданим шаблоном користувачеві буде видано інформаційне повідомлення «Для зазначеного виду заданий шаблон! Вкладення буде додано по цьому шаблону!» і виконана автоматична інтерпретація макропідстановок в шаблоні. В результаті додане вкладення буде виглядати як представлено на рисунку 4.12.

ДОГОВОР ПОСТАВКИ № 101-2014	
10.10.2014 г.	
Открытое акционерное общество «Рога и копыта» в лице Иванова Ивана Ивановича, действующего на основании статута, с одной стороны и «ОАО Восход», зарегистрированное и действующее в соответствии с законодательством № 987, именуемое в дальнейшем «Покупатель», в лице Петрова Николая Викторовича, действующего на основании статута, с другой стороны, при совместном упоминании «Стороны», заключили настоящий договор поставки (далее по тексту «договор»), на нижеследующих условиях:	

Рисунок 4.9 – Вкладения по шаблону

4.4.6 Розділи документів

Розділи документів (рисунок 4.13) використовуються для угруповання однотипних вкладень (договору, векселі, довідкові документи, конструкторська документація і т. д.).

[DRR] Разделы документов							
	Код ^	Наименование	Сокр.наим	Пр	Модули	Сист	Козфф.2
	01	Картинки	Картинки				
	1	Конструкторские документы	КонстрДок			<input type="checkbox"/>	
	ДГ	Договоры	Договоры			<input type="checkbox"/>	
	ДЛ	Документы делопроизводства	Делопроеизв			<input type="checkbox"/>	
	МЦ	Материальные ценности	Материалы			<input type="checkbox"/>	
	ПР	Приказы	Приказы			<input type="checkbox"/>	
	Ф	Фотокарточка	Фото			<input type="checkbox"/>	

Рисунок 4.10 – Розділи документів

4.4.7 Налаштування доступу до управління контентом

Для можливості налаштування видів, статусів, прав доступу для вкладень необхідно дозволити доступ до функції «Конструктор управління контентом». Доступ до цієї функції визначається за значенням групового параметра "Налаштування доступу до конструктора управління контентом" (DORA). Щоб користувачу доступна можливість створювати свої види вкладень та інше необхідно встановити режим = «Коригування» (рисунок 4.14).

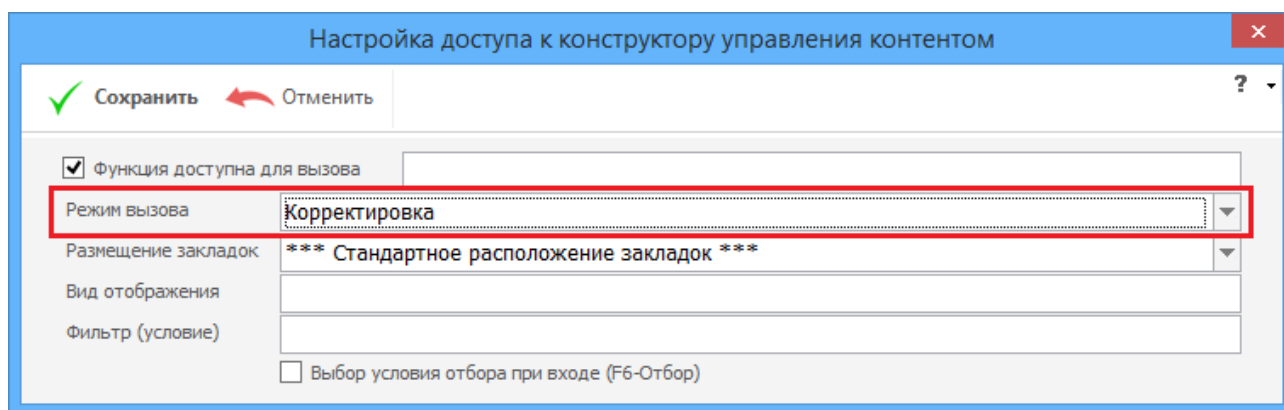


Рисунок 4.11 – Налаштування доступу до функції «Конструктор управління контентом»

Налаштування описаних групових параметрів для всіх груп можна виконати в функції " налаштування доступу "(Адміністрування системи → управління контентом → управління контентом → налаштування доступу).

4.4.8 Режими роботи з вкладеннями

При роботі з вкладеннями доступні наступні режими:

- додавання нового вкладення – у випадяючому списку відображаються всі доступні для додавання види;
- видалення вкладення – при роботі з вкладеннями, що зберігаються в окремих файлах (залежить від виду вкладення), автоматично буде запропоновано видалити сам вкладеного файлу з диска;
- перегляд і коригування вкладення відповідним переглядачем – файл вкладення (з диска або з бази даних) копіюється в тимчасовий каталог і відкривається на перегляд або коригування, якщо у користувача немає прав на коригування, то вкладення буде доступно тільки для перегляду.

Після перегляду тимчасовий файл видаляється. Отже, навіть якщо користувач внесе якісь зміни на даний файл (наприклад, файл на перегляд був відкритий в MS Word), то «вихідний файл залишиться незмінним. Якщо необхідно

зберегти зміни, внесені в режимі перегляду, необхідно зберегти файл під іншим ім'ям.

При коригуванні після збереження всіх необхідних змін у Тимчасовому файлі оновлена інформація буде перенесена в систему (в базу даних або в файл). Потім тимчасовий файл буде видалено.

Завантажити – можливість зберегти файл вкладення на локальному комп'ютері. При використанні даного режиму користувачеві буде запропоновано вибрати папку для збереження вкладення.

Сканувати – використовується для автоматичного сканування паперового документа і приміщення його електронної скан-копії вкладення.

Підписати / видалити ЕЦП – користувач може підписати вкладення за допомогою електронного підпису; в разі зміни вмісту вкладення підпис стане не валідною. Інформація про ЕЦП (стан, ПІБ і дата), відображається в окремій закладці.

Нова версія – можливість створити нову версію файлу вкладення. Всі доступні версії вкладення відображаються на окремій закладці «Версії». Одна з версій вкладення завжди є активною (активна версія відображається в загальному списку всіх вкладень).

Зауваження – можливість додати до файлу вкладення графічне зауваження. Всі зауваження до файлу вкладення відображаються на окремій закладці «Зауваження».

Додавання простого вкладення доступно, якщо інтерфейс був викликаний в режимі «Коригування». У режимі Перегляду» додавання неможливо.

Можливість додавання вкладення з заданим видом визначається настройками доступу. Для програмного виклику додавання вкладення в C # коді необхідно використовувати метод: `Originals.AddOriginal ("таблиця", "вид вкладень", "значення ключа", "повний шлях до файлу з відносно сервера програми")`.

Для програмного виклику додавання простого вкладення в FoxPro коді необхідно використовувати функцію `smpdor_action ()` в режимі "NEW". Додаткові параметри функції `smpdor_action ()` див. в API.

Видалення простого вкладення доступно, якщо інтерфейс був викликаний в режимі «Коригування». У режимі Перегляду» видалення неможливо. Допускається видалення тільки «своїх» вкладень – тих, що були додані поточним користувачем. Для адміністратора системи (право «Адміністратор», встановлене в довіднику користувачів системи) доступно видалення, у тому числі вкладень, створених іншими користувачами.

Можливість видалення вкладення з заданим видом визначається настройками доступу. Для програмного виклику видалення вкладення в C # коді необхідно використовувати метод: `Originals.DeleteOriginal ("таблиця", "значення ключа", 1)`.

Для програмного виклику видалення простого вкладення в FoxPro коді необхідно використовувати функцію `smpdor_action()` в режимі "DEL". Додаткові параметри функції `smpdor_action ()` див. в API.

Якщо потрібно заборонити видалення вкладень з бази даних, то необхідно налаштувати глобальний параметр «Налаштування вкладень» (GLDOR) - вказати Дію при видалення = Скасувати. При такій настройці при видаленні вкладення рядок не буде видалена, а буде позначена як скасована. Можливість переглядати і видаляти скасовані вкладення настраюється інтерфейс меню «Налаштування роботи з контентом» (рисунок 4.15).

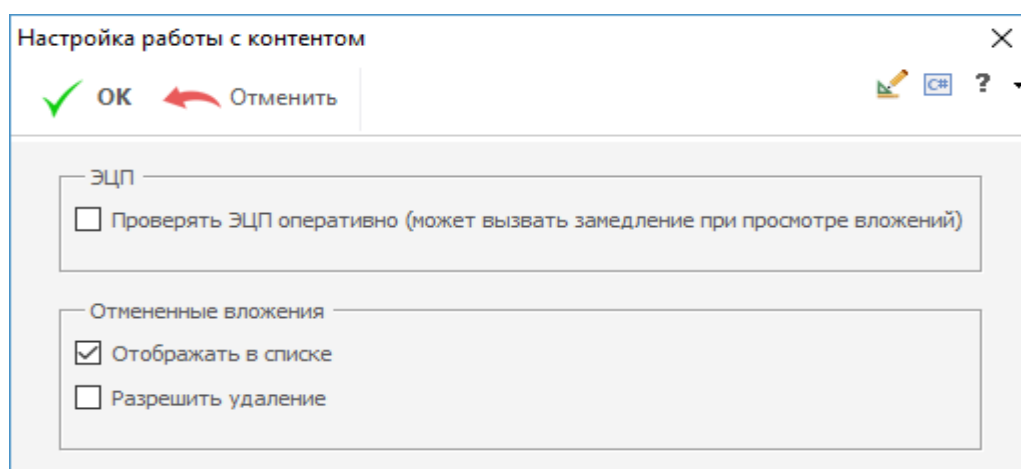


Рисунок 4.12 – Параметр "налаштування роботи з контентом". Доступ до відмінених вкладень

4.4.9 Перегляд і коригування вкладки

При перегляді файл буде завантажений на комп'ютер користувача в тимчасову папку. Після завантаження буде викликано додаток, асоційоване на комп'ютері з файлами вибраного типу (наприклад, для файлів типу .DOC буде викликаний Microsoft Word і т. д.). Після завершення перегляду документа файл видаляється з тимчасової папки.

Користувач при перегляді вкладки має можливість створити копію документа і зберегти її на своєму комп'ютері або на мережевому ресурсі. Але ніякі зміни документа на комп'ютері користувача не потраплять в базу даних. Якщо необхідно змінити вкладення, то необхідно додати нове вкладення і видалити старий.

Режим перегляду простих вкладень доступний завжди, незалежно від режиму виклику основного інтерфейсу (перегляд або коригування). Для перегляду доступні всі прості вкладення, незалежно від того, ким вони були додані.

Коригування простого вкладення доступна тільки користувачеві-власнику, який додав вкладення, або адміністратору системи IT-Enterprise.

Режим перегляду і коригування вкладень із заданими видами визначається настройками доступу.

Для програмного виклику перегляду вкладення в C # коді необхідно використовувати метод: `Originals.ViewOriginal ("таблиця", "значення ключа", 1)`.

Для програмного виклику перегляду простого вкладення в FoxPro коді необхідно використовувати функцію `smpdor_action()` в режимі "VIEW". Додаткові параметри функції `smpdor_action ()` див. в API.

4.4.10 Збереження вкладки

Для збереження файлу вкладення на локальному диску комп'ютера або на мережевому ресурсі необхідно натиснути на кнопку . Користувачеві буде виданий діалог для вибору місця збереження файлу на комп'ютері (рисунки 4.16)

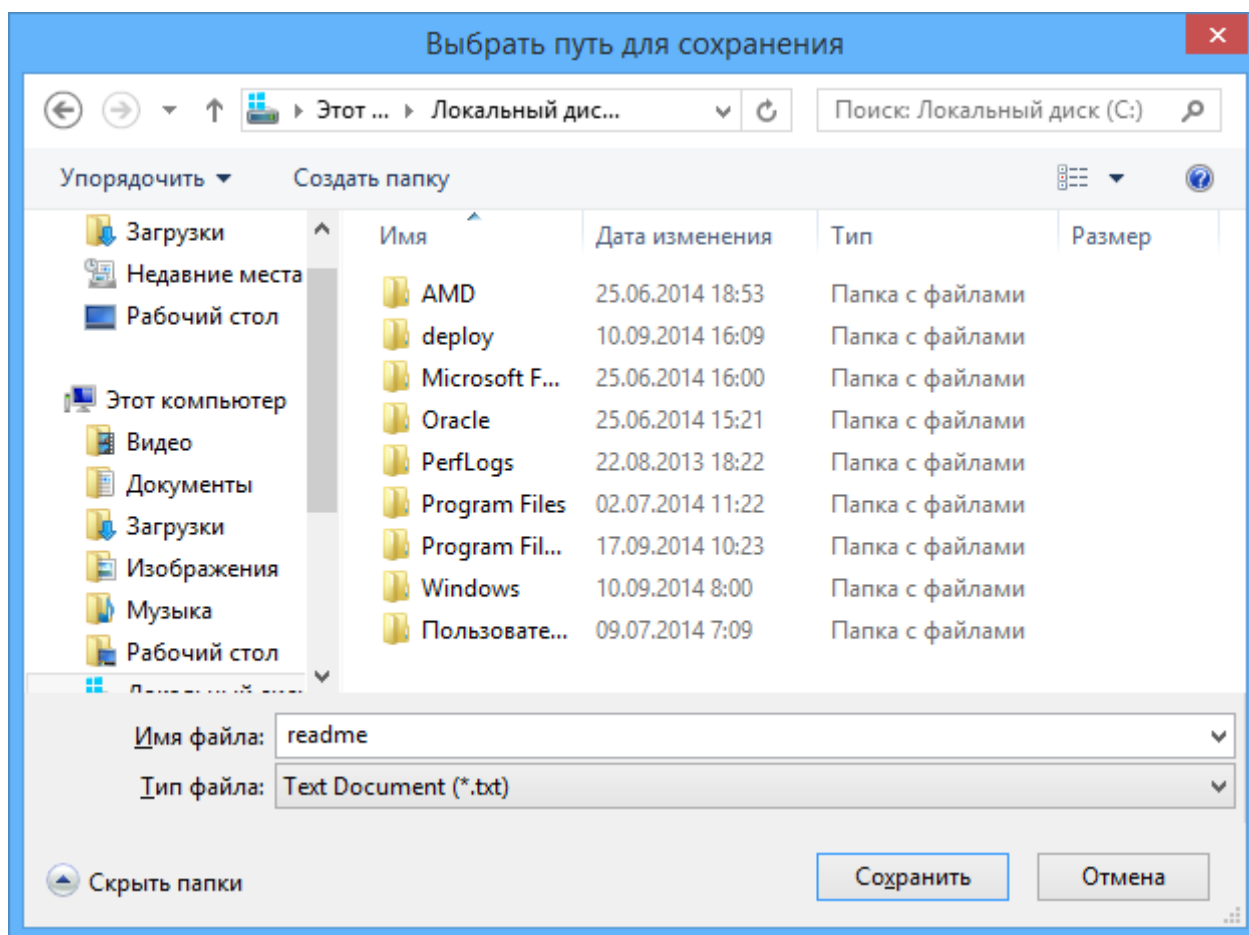


Рисунок 4.13 – Діалог для збереження файлу вкладення

Режим збереження вкладення доступний завжди, незалежно від режиму виклику основного інтерфейсу (перегляд або коригування) та прав доступу. Для збереження доступні всі вкладення, незалежно від того, ким вони були додані.

Для збереження декількох файлів вкладень у вигляді архіву доступний режим "завантажити в архіві". За замовчуванням в архів поміщаються всі вкладення, у користувача є можливість вибрати потрібні (відмітка множинного вибору).

Для програмного виклику збереження вкладення в C # коді необхідно використовувати метод: `Originals.SaveOriginal ("таблиця", "значення ключа", 1)`.

Для програмного виклику збереження простого вкладення в FoxPro коді необхідно використовувати функцію `smpdor_action()` в режимі "SAVE". Додаткові параметри функції `smpdor_action ()` див. в API.

Якщо існує паперовий документ, який необхідно помістити у вкладення, то для цього можна використовувати кнопку «Сканувати». Користувачеві буде виданий діалог для завдання параметрів сканування, після сканування отриманий файл буде автоматично додано до вкладень.

Для вже доданих вкладень у форматі PDF існує можливість досканувати сторінки в раніше доданий файл. Для цього використовується режим «Сканувати з додаванням».

4.5 ЕЦП в додатках

Будь-який документ, поміщений у вкладення, може бути підписаний ЕЦП користувача. Наявність коректної ЕЦП гарантує, що, саме цей користувач запевнив саме цей документ саме в цей час. Всі підписи вкладення відображаються на окремій закладці «ЕЦП» (рисунок 4.17).

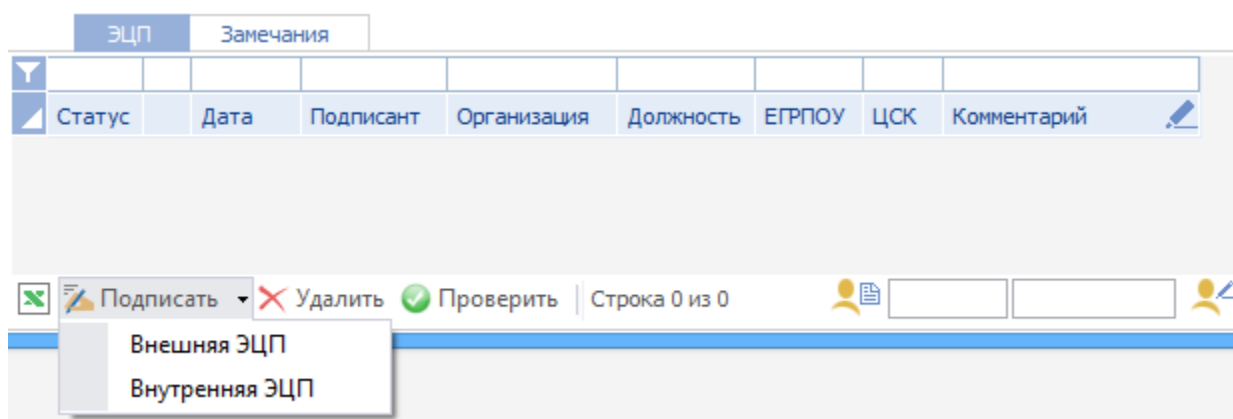


Рисунок 4.14 – Вкладка з ЕЦП

В системі підтримується можливість використання юридично-значущого цифрового підпису (зовнішня ЕЦП) і внутрішнього підпису IT-Enterprise. Більш детально про ЕЦП див. у розділі «Електронний цифровий підпис».

За замовчуванням валідація коректності підписів не виконується, оскільки перевірка зовнішньої ЕЦП може займати тривалий час. Для перевірки підпису необхідно використовувати кнопку «Перевірити». Але якщо користувачеві

необхідно перевіряти підписи завжди оперативно, то цю можливість можна включити в користувацькому параметрі «Настройка работы с контентом» (рисунок 4.18).

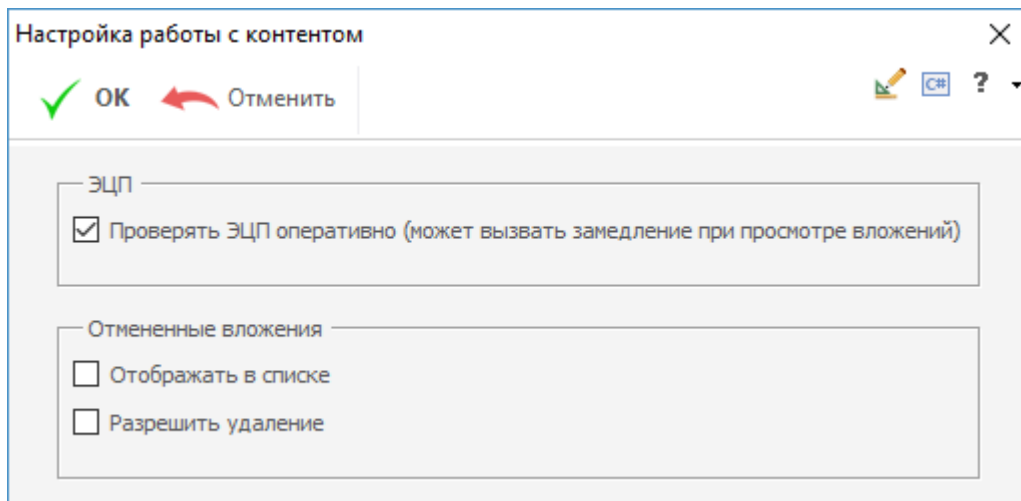


Рисунок 4.15 – Параметр "налаштування роботи з контентом". Перевірка ЕЦП

Для програмного виклику підпису вкладення за допомогою ЕЦП в С # коді необхідно використовувати метод: `Originals.SignOriginal ("таблиця", "значення ключа", 1)`.

Для програмного виклику видалення ЕЦП підпису вкладення в С # коді необхідно використовувати метод: `Originals.DeleteOriginalSignature ("таблиця", "значення ключа", 1)`.

Для програмного виклику підпису простого вкладення з допомогою ЕЦП в FoxPro коді необхідно використовувати функцію `smpdor_action()` в режимі "SIGNATURE", для видалення підпису в режимі "SIGNDEL". Додаткові параметри функції `smpdor_action ()`.

Для вкладень підтримується версійність. Для будь-якого вкладення можна створити кілька версій. Одна з версій вкладення буде активною (використовуватися за замовчуванням при роботі). Всі доступні версії вкладень відображаються на закладці «Версії» (рисунок 4.19).

Вложения		Версии			
Версия	Комментарий	Размер (байт)	Автор	Дата создания	
1	S_PJZ_TA01	33 792			
2	v1.1	22 528		17.10.2015 17:52	

Рисунок 4.16 – Вкладка з версіями

Для додавання нової версії можна використовувати такі режими:

- створення нової версії з існуючого вкладки;
- додавання нової версії з готового файлу;
- отримання нової версії шляхом сканування.

В результаті використання одного з перерахованих режимів додана версія автоматично стає активною. Активну версію можна поміняти з використанням кнопки "зробити версію активною". При наявності відповідного доступу версії вкладень можна коригувати і видаляти.

Для версій вкладень у форматі docx доступна можливість порівняння засобами MS Office Word. Для використання необхідна наявність встановленого ПЗ на ПК користувача.

Для вкладень існує можливість додавати довільні графічні зауваження. Для додавання зауважень використовується Adobe Reader. Всі графічні зауваження оформляються у форматі PDF. Якщо для вкладення є зауваження, то вони відображаються на окремій закладці «Зауваження» (рисунок 4.20).

ЭЦП		Замечания		
Описание замечания	Устранено	Когда устранено	Кто устранил	
Замечания от 12.10.2016 15:05	<input type="checkbox"/>			

Рисунок 4.17 – Вкладка з зауваженнями

Для зауважень можна додати текстовий опис. Для позначки після усунення зауваження використовується ознака "усунуто", також відображається коли і хто допрацював зауваження.

Ескізи використовуються для зберігання файлу-копії вкладення у спрощеному форматі (наприклад, для складних графічних файлів в якості ескізу використовується файл у вигляді картинки). Якщо для вкладення існує ескіз, то саме ескіз буде відображатися на закладці перегляду вкладення. Ескізи доступні тільки для вкладень із заданими видами, які використовуються в інтерфейсах версії 10. Для створення ескізу необхідно перевизначити метод `GetDraftFile` в бізнес-логіці виду вкладення.

Вкладення можна зашифрувати з використанням ключа ЕЦП. Для цього використовується режим «Зашифрувати». Після введення пароля зашифрований файл можна зберегти на ПК користувача або відправити поштою.

Прості вкладення доступні для таблиць, у яких визначено первинний ключ.

Вкладення із заданими видами доступні при наявності опису таблиці прив'язки вкладень.

Слід врахувати, що робота з файлами великих розмірів може сильно збільшити розмір бази даних, а також завантажити мережу, особливо, якщо доступ до сервера додатків здійснюється через Інтернет. Не рекомендується використовувати інструмент вкладень для зберігання файлів великого розміру в базі даних.

Для обмеження розміру вкладень, які можуть додаватися користувачами (для будь-яких вкладень по всіх таблиць), використовується глобальний параметр «Налаштування вкладень» (рисунок 4.21). У параметрі можна задати допустимий розмір в Кб.

Іноді необхідно не видаляти вкладення, а приховати його від використання (скасувати). Для цього використовується настройка "дія при видаленні". Якщо в налаштуванні вказано "скасувати", то при видаленні будь-якого вкладення рядок не буде видалена з бази, а буде позначена як скасована. Адміністратору буде доступна можливість відновити вкладення.

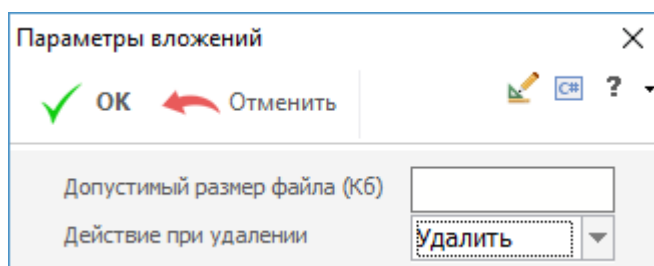


Рисунок 4.18 – Настойка допустимого розміру вкладень

4.6 Реалізація управління користувачами

Розглянемо форму управління користувачами (рисунок 4.22):

- показувати значення станом на – дозволяє переглянути і задавати параметри користувача станом на зазначену дату;
- підключений – дозволяє користувачеві працювати з сервісом;
- обліковий запис – запис користувача з довідника користувачів;
- параметри користувача.

Перелік параметрів користувача залежить від налаштувань сервісу. Параметри користувача, які за умовчанням є в кожному сервісі:

- перегляд фін. інформації – якщо відзначений, то користувач буде включений в розсилку рахунків, зможе управляти підпискою організації і бачити фінансову статистику організації в особистому кабінеті;
- адміністратор сервісу - якщо відзначений, то користувач зможе управляти обліковими записами своєї організації через особистий кабінет.

Система управління сервісами дозволяє описати права користувачів на кожному сервісі за допомогою механізму «параметрів користувача».

Для кожного сервісу може бути описаний свій набір параметрів. Значення параметрів індивідуальні для кожного користувача і можуть бути змінені адміністратором.

Всі зміни параметрів користувача передаються на комплект сервісу для застосування.

Рисунок 4.19 – Форма налаштування користувача

В системі користувачі можуть прив'язуватись до організації, це дозволяє реалізувати механізм управління користувачами адміністратором організації.

Основні налаштування організації (рисунок 4.23):

- організація – організація, для якої надається доступ;
- підключений – ознака підключення організації до сервісу, необхідно включити, при цьому з'являться додаткові вкладки для налаштування параметрів підписки організації на послугу;
- дата початку роботи – дата, з якої починається тарифікація організації, використовується для послуг з типом тарифікації «за період»;
- ручний облік – ознака, при включенні якого документообіг по організації не буде проходити через фінансову систему (рахунки створюються без ПДВ з ознакою нал, акти не створюються);
- знижка,% – відсоток знижки для організації;
- договір на надання послуги – реквізити договору, згідно з яким організації надається доступ до сервісу;
- комплект – ідентифікатор комплекту сервісу підприємства. заповнюється якщо даного сервісу використовується індивідуальний комплект для кожного підприємства;

- основна мова спілкування – мова, на якому будуть відправлятися поштові повідомлення на адреси організації;
- дозволити роботу в кредит до – дата, до якої дозволена робота в кредит, в незалежності від стану рахунку підприємства (використовується для послуг з типом тарифікації «за період»).

Рисунок 4.20 – Форма налаштування організації

Додаткові налаштування (рисунок 4.24) – додаткові опції для тонкої настройки механізму автоматичного створення фінансових документів:

- включати суму заборгованості в автоматично формуються рахунки – при виставленні нового рахунку включати в загальну суму рахунку крім передоплати суми всіх минулих рахунків, які не були оплачені;
- завжди виставляти рахунок з датою дії рівній останнього дня місяця – за замовчуванням при тарифікації за період нові рахунки виставляються на місяць або 30 днів вперед від дати закінчення грошей на рахунку підприємства. Ця опція дозволяє округляти дату, до якої нараховується сума передоплати, до кінця місяця;

– при надходженні платежів списувати грошові надходження на минулі неоплачені періоди – у разі, якщо на рахунку організації закінчилися гроші, керуючий комплект автоматично генерує сигнал про відключення послуги, який може бути оброблений сервісом. При надходженні грошей на рахунок організації автоматично генерується сигнал про включення послуг сервісу. Якщо включена ця опція, то гроші за користування послугою будуть списані в тому числі і за період, коли на рахунку організації були відсутні грошові кошти та сервіс був відключений;

– формула перекладу ціни в валюту рахунку - дозволяє описувати правила перекладу суми документів, створених на керуючому комплекті, в реальну валюту, в якій виставляються рахунки. Для використання необхідно, щоб фінансова система підтримувала виконання скриптів на мові IronPython.

☐ Включать сумму задолженности в автоматически формируемые счета

☐ Всегда выставлять счета с датой действия равной последнему дню месяца

☐ При поступлении платежей списывать денежные поступления на прошлые неоплаченные периоды

Формула перевода цены
в валюту счета

Рисунок 4.21 – Додаткові налаштування організації

На вкладці «Встановлювані користувачем» (рисунок 4.25) здійснюється підписка на послуги і настройка параметрів послуг для організації.

Показувати значення станом на – дата, з якої почнуть діяти настройки організації. Заповнюється, якщо необхідно надати доступ організації до сервісу після певної дати (наприклад, з 1 числа наступного місяця). Якщо дата запуску не

визначена, можна залишити поточну дату. У будь-якому випадку сервіс буде доступний для використання тільки після оплати.

Показывать значения по состоянию на 25.07.2016	
<input checked="" type="checkbox"/> Один конкурентный доступ	40, USD
Количество	1
— Приложения —	
<input type="checkbox"/> Пакет продажи, производство, закупки, затраты	219, USD
<input type="checkbox"/> Пакет управление персоналом	119, USD
— Продажи —	
<input type="checkbox"/> Управление взаимоотношениями с клиентами (CRM)	20, USD
<input type="checkbox"/> Управление заказами	20, USD
<input type="checkbox"/> Управление договорами, счетами	30, USD
<input type="checkbox"/> Взаиморасчеты с клиентами	20, USD
<input type="checkbox"/> Публикация остатков на сайте	30, USD

Рисунок 4.22 – Форма налаштування параметрів послуги

На вкладці перераховані всі послуги сервісу. Для настройки підписки на послугу необхідно:

- включити прапорці навпроти тих послуг, які будуть доступні організації (якщо для підключається послуги описані залежні послуги, то вони будуть підключені автоматично);
- якщо у послуги є параметри (наприклад, кількість), то вказати необхідне значення.

4.7 Реалізація обмін інформацією між компонентами системи та арі

Система IT-Enterprise надає набір різних засобів для програмного обміну даними з іншими програмними продуктами.

Одним з таких засобів є Web-сервіси – можливість публікації виклику розрахунків IT-Enterprise для отримання даних з IT-Enterprise, введення даних в IT-Enterprise, отримання аналітичних звітів і зведень і т. д.

Звичайно Web-сервіси використовуються для інтеграції з іншими програмними продуктами, з їх допомогою стороннє ПЗ має можливість звертатися до даних, причому доступ до даних може надаватися віддалено (через інтернет), і тільки санкціонований.

Перелік доступних розрахунків, опублікованих у вигляді web-сервісів, визначає адміністратор системи IT-Enterprise.

Зовнішніми додатками можуть виступати як Web-додатки, Windows-додатки, так і додатки для мобільних платформ – native-програми під Android, iOS, Windows Phone і т. д.

Схема взаємодії представлена на рисунку 4.26.

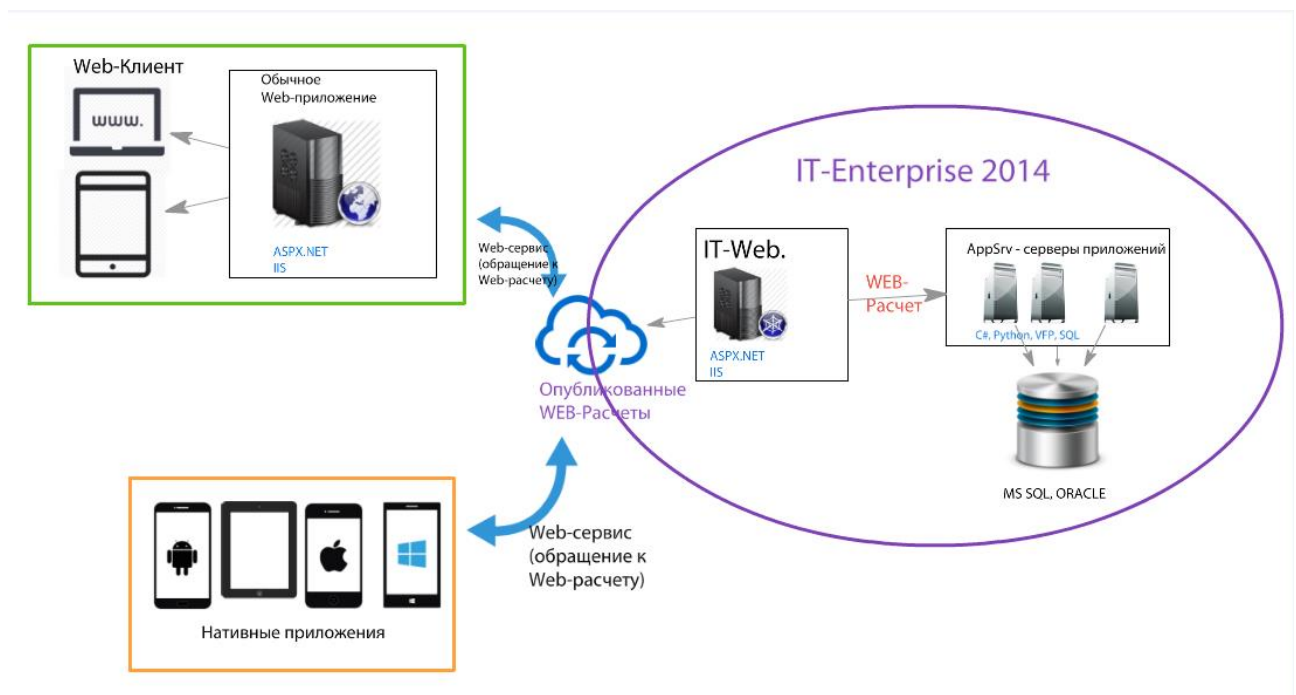


Рисунок 4.23 – Схема взаємодії IT-Enterprise з зовнішніми системами

4.7.1 Web-розрахунки

Web-розрахунки – це функції, реалізовані всередині системи IT-Enterprise стандартними засобами розробки системи (на мовах C#, Python, VFP), які приймають на вхід запити від зовнішніх додатків (параметри) і виконують які-небудь функціональні дії в системі IT-Enterprise.

Наприклад:

- перегляд поточних залишків на заданому складі ДП;
- резервування товару;
- додавання заявки на закупівлю.

Web-розрахунки знаходяться всередині IT-Enterprise, і всі дії з даними IT-Enterprise виконуються всередині самої системи, а зовнішні програми не мають прямого доступу до даних, і "спілкуються" з IT-Enterprise тільки шляхом "запит-відповідь" (через опубліковані Web-розрахунки).

Web-розрахунки універсальні. Не важливо, як реалізовано зовнішню програму – це класичний Windows або Web-клієнт або ж нативне додаток для Android – робота з IT-Enterprise виконується з допомогою єдиного опублікованого Web-розрахунку. Це дозволяє один раз описавши бізнес-логіку в Web-розрахунку на стороні зовнішнього клієнта виконувати тільки візуалізацію даних.

Передналаштовані Web-розрахунки поставляються розробником IT-Enterprise і можуть бути використані безпосередньо або лягати в основу користувача Web-розрахунків, створюються користувачами системи.

Підтримка розробника IT-Enterprise – нові системні Web-розрахунки поставляються в рамках оновлень системи.

Web-розрахунки реалізуються стандартними засобами розробки системи IT-Enterprise (C#, Python, VFP), їх конфігурування може виконувати відділ супроводження системи, без залучення сторонніх програмістів з підрядних організацій.

Публікація Web-розрахунків дозволяє викликати даний розрахунок сторонньому ПЗ. Доступ до розрахунку (дозволений чи ні) задається адміністратором системи.

Розробка web-розрахунків виконується у функції "Web-розрахунки" (адміністрування – бібліотеки коду – Web-розрахунки).

Екран коригування web-розрахунку (рисунок 4.27).

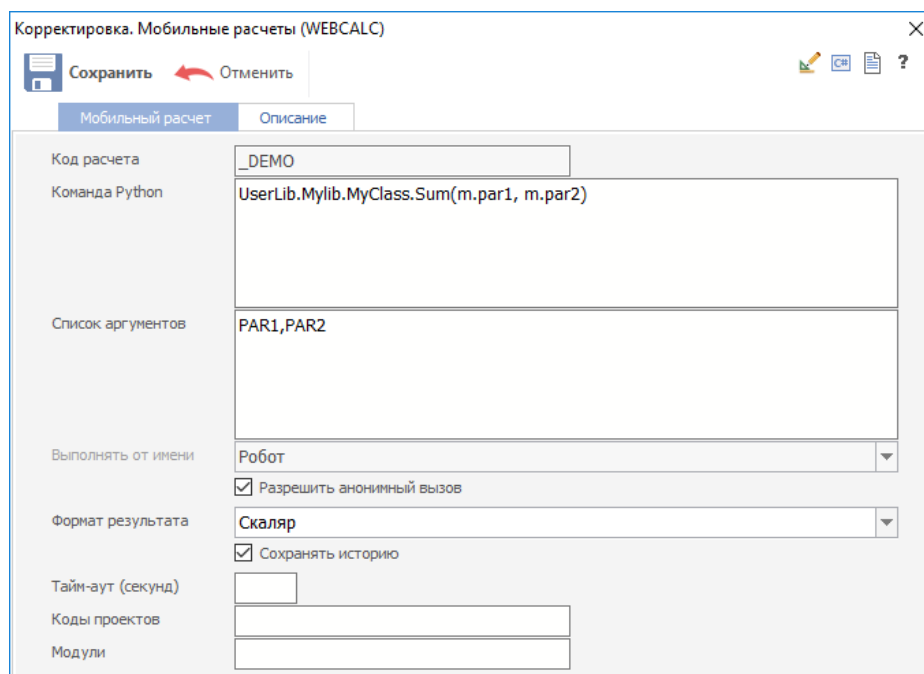


Рисунок 4.24 – Екран коригування web-розрахунку

Код розрахунку – унікальний ідентифікатор розрахунку. Команда Python – програма на мові Python, яка буде виконана при виклику web-розрахунку. Мова Python є гнучким універсальним механізмом, за допомогою якого можна використовувати будь-яку функціональність IT-Enterprise. Список аргументів – список імен параметрів, прийнятих web-розрахунком. Значення зазначених параметрів передаються з параметрів виклику web-розрахунку в команду Python. Дозволити анонімний виклик – вказує, чи можна виконувати розрахунок користувачам, які не пройшли процедуру аутентифікації і не отримали тимчасовий «квиток». Виконувати від імені – дана ознака вказує під яким обліковим записом IT-Enterprise потрібно виконувати розрахунок:

- під обліковим записом робота (без імперсоналізації);
- під обліковим записом користувача, що викликав розрахунок;

- під обліковим записом користувача, включаючи SQL-підключення.

Використання імперсоналізації займає деякий час, що знижує швидкість відгуку системи. Якщо результат розрахунку однаковий для всіх користувачів, то імперсоналізацію краще не використовувати. Якщо розрахунок поводитьися по-різному для різних користувачів, то використовувати імперсоналізацію.

Формат результату (рисунок 4.28) – вказує, в якому вигляді розрахунок повертає свій результат викликає стороні:

- скаляр: просте значення у вигляді рядка;
- колекція об'єктів: перелік значень рядків, чисел, дат);
- словник об'єктів: перелік значень «ключ»-«значення»;
- ім'я таблиці в локальному sql;
- набір таблиць в локальному sql (імена через кому);
- екземпляр будь-якого публічного класу, анонімний об'єкт;

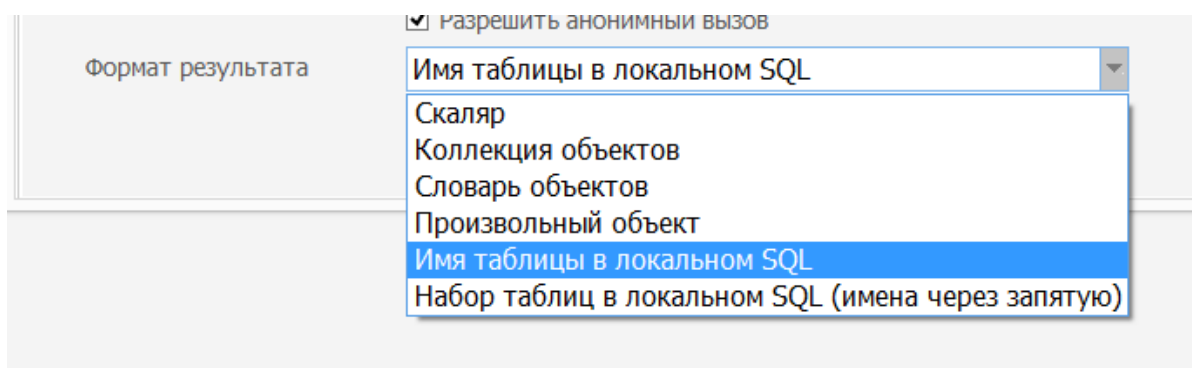


Рисунок 4.25 – Формати результату виконання web-розрахунків

Опис: текстовий опис розрахунку. У ньому має бути зазначено: призначення розрахунку, опис вхідних параметрів (їх призначення, типи, особливості використання), опис повернутого результату.

Передбачена можливість налаштування доступу користувачів до виклику web-розрахунків. Для кожного розрахунку можна вказати перелік користувачів або груп, яким дозволено або заборонено викликати розрахунок.

Налаштування доступу виконується в інтерфейсі «Web-розрахунки». Для установки досить натиснути кнопку «Налаштувати доступ» і відзначити галочками користувачів або групи, яким можна або не можна викликати розрахунок (рисунок 4.29).

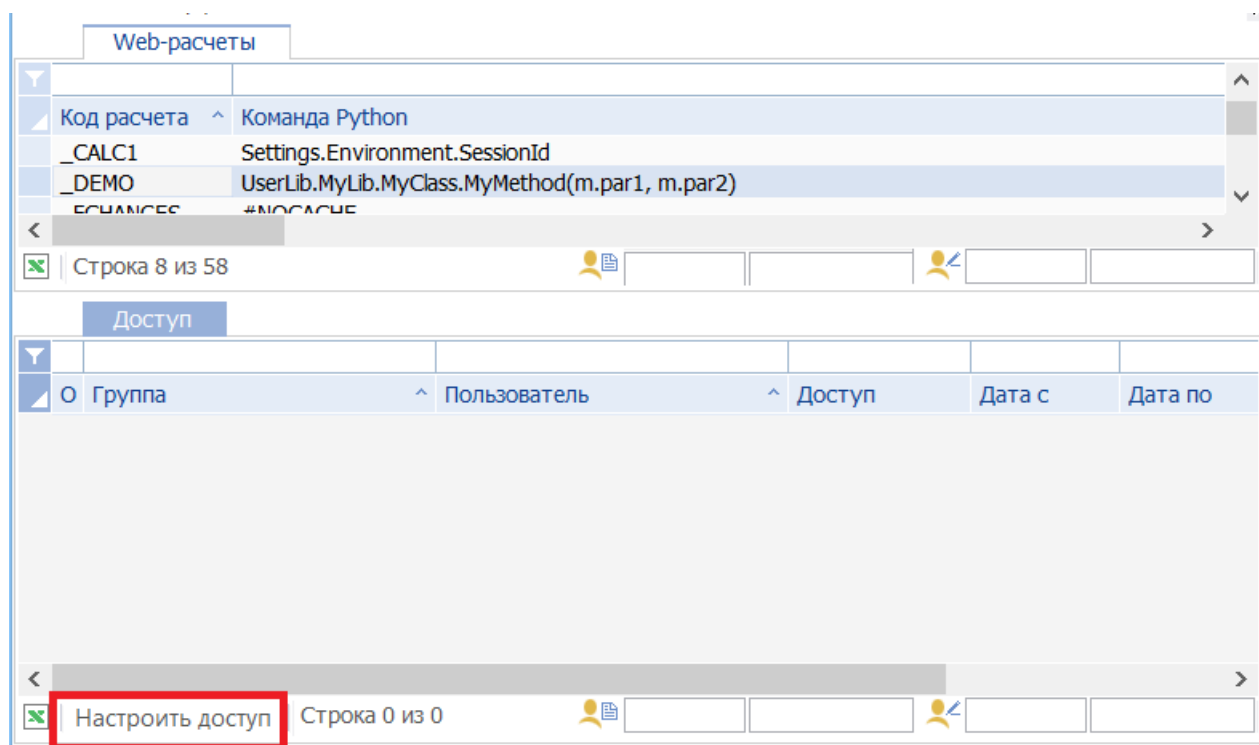


Рисунок 4.26 – Настройка доступа

4.7.2 Виклик web-сервісів

Допускається як авторизований доступ до web-розрахунків IT-Enterprise, так і анонімний доступ.

Авторизований доступ – для доступу до web-сервісів користувач повинен пройти перевірку логіна і пароля. Використовується ті ж логін і пароль, які служать для входу в звичайний клієнт IT-Enterprise. Авторизований користувач має доступ до всіх web-розрахунків, крім тих, які йому явно заборонені налаштуванням доступу. Використовується в ситуаціях, коли користувачі додатки відомі, для них заведені облікові записи в IT-Enterprise. Тобто, користувачами програми є

обмежене коло контрагентів (персонал, постачальники, авторизовані клієнти і т. п.).

Анонімний доступ – для доступу до web-сервісів не потрібно проходити перевірку логіна і пароля, доступні тільки ті web-розрахунки, для яких встановлено ознаку "дозволити анонімний виклик «і для яких налаштований доступ у вигляді» всім дозволено". Використовується в ситуаціях, коли потрібно надати доступ до інформації з IT-Enterprise великій кількості невідомих користувачів (наприклад, відвідувачам сайту підприємства).

4.7.3 Формат обміну даними

Для обміну даними між стороннім додатком і веб-сервісами IT-підприємство використовує формат даних JSON.

Це загальновідомий універсальний формат. Розмір даних: дані у форматі JSON і мають мінімальний розмір, особливо в порівнянні з форматом XML-файлі. Розмір даних впливає на розмір трафіку між клієнтським додатком і веб-сервісами IT-підприємство. Швидкість обробки: обробка даних у форматі JSON і виконується швидко. Під обробкою мається на увазі упаковка структурованих даних в рядок у форматі JSON і зворотне перетворення. Кроссплатформенність: будь-які платформи і мови програмування мають засоби для роботи з JSON B (C#, Java, iOS, андроїд, скрипти,...).

4.7.4 Основні принципи взаємодії з web-сервісами з клієнтського додатка

Якщо додаток використовує авторизований доступ, то першим зверненням до web-сервісу повинна бути перевірка логіна і пароля користувача (додаток повинен попередньо запитати у користувача логін і пароль).

Для перевірки логіна і пароля використовується web-метод LoginEx(string login, string password). Даний метод повертає "квиток" - секретний рядок, який підтверджує, що користувач пройшов аутентифікацію. Даний «квиток» потрібно

передавати в усі наступні виклики web-розрахунків для підтвердження аутентифікації користувача.

Вхідні данні: login – логін користувача (рядок) та password – пароль користувача (рядок). Вихідні: інформація про успішність аутентифікації (рядок json) у вигляді:

```
{
  "Success": true,
  "UserName": "ім'я користувача",
  "Ticket": "тікет",
  "FailReason": null,
  "NeedChangePassword": false,
  "TempPasswordRequired": false,
  "TempPasswordMessage": null
}
```

Де: Success – успішність аутентифікації (true/false), UserName – ПІБ користувача, Ticket – тимчасовий квиток, FailReason-причина неуспішної аутентифікації, NeedChangePassword – ознака необхідності змінити пароль, TempPasswordRequired – ознака необхідності введення тимчасового пароля (використовується при двофакторної аутентифікації), TempPasswordMessage-повідомлення для користувача про необхідність ввести часовий пароль.

Термін дії квитка – 1 година. Якщо користувач не звертався до web-сервісу більше години, то його квиток застаріває. При спробі викликати web-розрахунок із застарілим квитком буде видано код повернення "WRONG_TICKET". При отриманні такого коду повернення необхідно пройти аутентифікацію повторно (повторно викликати LoginEx і отримати новий квиток).

Для інтернет-розрахунків виклику використовується веб-метод ExecuteEx, який має наступні параметри: calcId – ідентифікатор веб-розрахунку викликається (рядок), args – рядок параметрів у форматі JSON (має вигляд: {par1: 10, par2:"рядок"}), ticket – квиток, отриманий викликом методу LoginEx. Якщо

виконується анонімний виклик розрахунку, то в якості квитка подається порожній рядок.

Метод `ExecuteEx` повертає дані, які веб-розрахунок повернув, упаковані в формат даних JSON.

Приклад виклику веб-анонімного з розрахунку на JavaScript (за допомогою фреймворка jQuery з):

```
$.ajax({  
    url: "https://server.com/ws/webservice.asmx/ExecuteEx?pureJSON=",  
    type: "POST",  
    dataType: "json",  
    contentType: "application/json",  
    data: JSON.stringify({  
        calcId: "_SUM",  
        args: JSON.stringify({ val1: 10, val2: 20 }),  
        ticket: ""  
    }),  
    success: function(result){  
        alert(result);  
    },  
    error: function(e) {  
        alert("exception");  
    }  
})
```

4.8 Реалізація веб порталу

Клієнтський додаток реалізований з використанням технологій описаних в попередніх розділах: Asp.Net, Web-розрахунки та Vue.js. та надає можливість користувачам зовнішніх організацій працювати з документами, накладати ЕЦП, надавати доступ до документів іншим користувачам.

Веб сайт надає користувачу такі можливості (рисунок 4.30):

- реєстрація/Авторизація на порталі;
- налаштування профілю;
- завантаження документу до системи;
- перегляд документу;
- збереження документу;
- підписання документу;
- налаштування доступу до документу;
- видалення та відхилення документу.

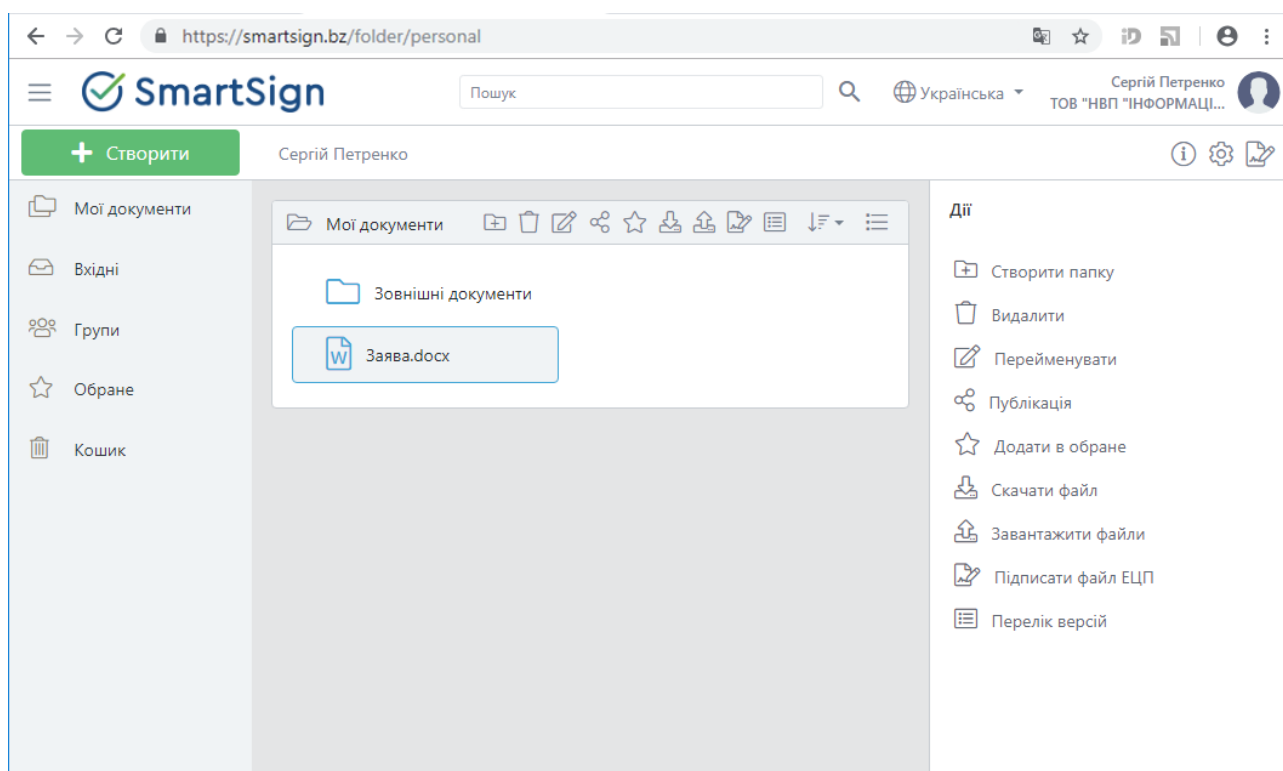


Рисунок 4.27 – Інтерфейс роботи з документом у веб порталі

4.9 Розробка структури БД

В ERP системі реалізовані засоби для розробки програмних рішень з одночасною підтримкою таких СУБД, як MS SQL Server та Oracle. Для цього

розроблені такі інструменти, як візуальний конструктор БД та будівники SLQ запитів на мові C# та IronPython.

Проектуючи БД приймалася до уваги нормалізація таблиць, а також, використовувались індекси для підвищення ефективності виконання запитів.

Нормальна форма – вимога, що пред'являється до відносин в теорії реляційних баз даних для усунення з бази надмірності, яка потенційно може привести до помилкових результатів вибірки або зміни даних.

Нормалізація - процес перетворення бази даних до вигляду, відповідальному нормальним формам. Нормалізація дозволяє убезпечити базу даних від логічних і структурних проблем, званих аномаліями даних.

Аномаліями називається така ситуація в БД, яка призводить до протиріччя у БД, або суттєво ускладнює обробку даних. Розрізняють аномалії модифікації, додавання і вилучення [27].

Таблиці в базі даних можуть мати велику кількість рядків, які зберігаються у довільному порядку, і їх пошук за заданим значенням шляхом послідовного перегляду таблиці рядок за рядком може займати багато часу. Індекс формується зі значень одного чи кількох стовпчиків таблиці і вказівників на відповідні рядки таблиці і, таким чином, дозволяє знаходити потрібний рядок за заданим значенням. Прискорення роботи з використанням індексів досягається в першу чергу за рахунок того, що індекс має структуру, що оптимізована для пошуку — наприклад, збалансованого дерева.

Деякі СУБД розширюють можливості індексів введенням можливості створення індексів за виразами. Наприклад, індекс може бути створений за виразом `upper(last_name)` і відповідно буде зберігати посилання, ключем яких будуть значення поля `last_name` в верхньому регістрі. Крім цього, індекси можуть бути оголошені як унікальні так і не унікальні. Унікальний індекс реалізує обмеження цілісності на таблиці, виключаючи можливість вставки значень, що повторюються.

Схема основної частини БД системи представлена на рисунку 4.31.

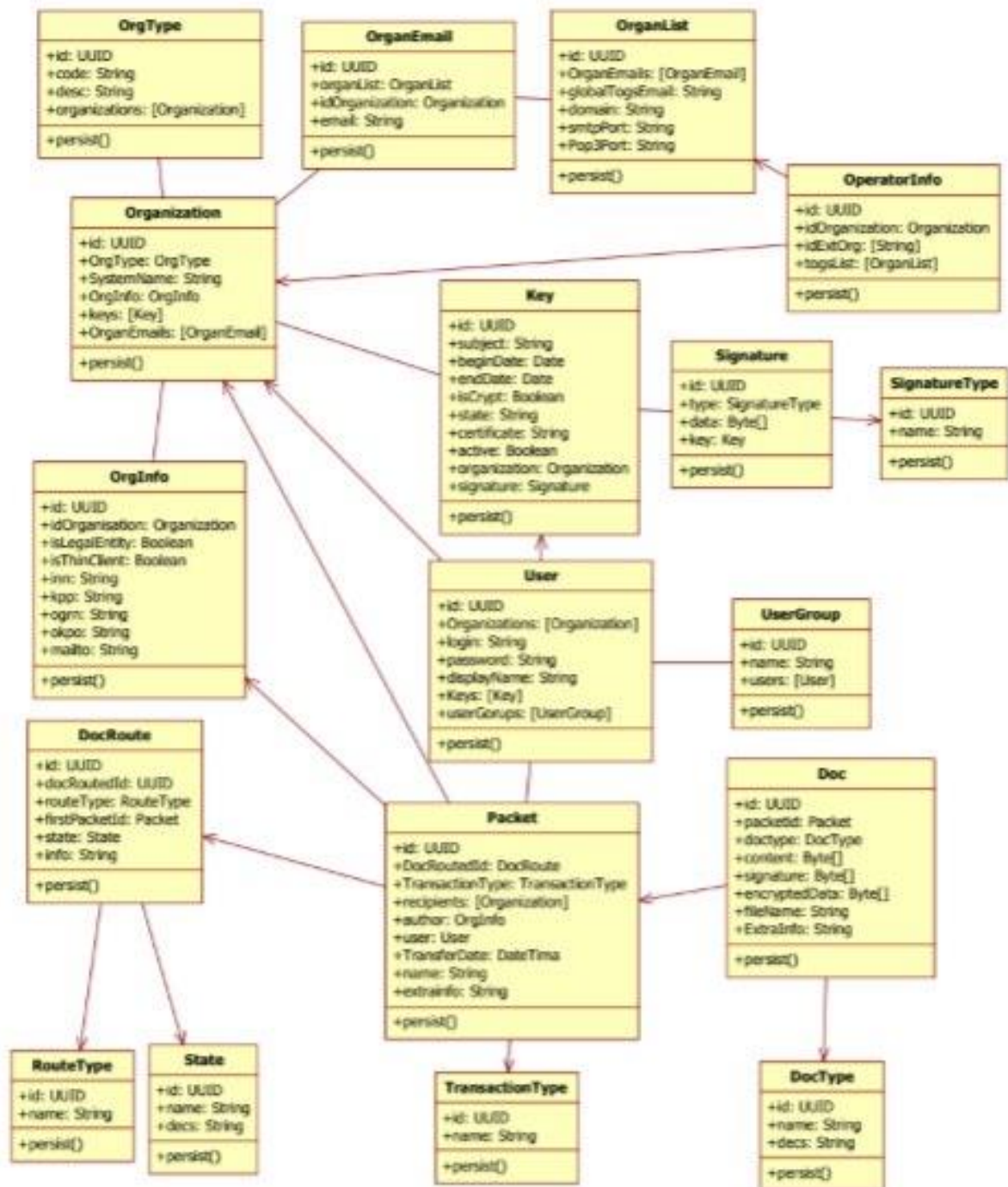


Рисунок 4.28 – Схема основної частини БД

5 ТЕСТУВАННЯ ПРОГРАМНОГО КОМПЛЕКСУ

Тестування програмного забезпечення – процес дослідження програмного забезпечення з метою отримання інформації про якість продукту.

Юніт-тест (модульне тестування) – процес в програмуванні, що дозволяє перевірити на коректність окремі модулі вихідного коду програми.

Рефакторинг – процес зміни внутрішньої структури програми, що не зачіпає її зовнішньої поведінки і має на меті полегшити розуміння її роботи. В основі рефакторинга лежить послідовність невеликих еквівалентних (тобто зберігають поведінку) перетворень [26].

Інтеграційний тестування (інтеграційний тестування.) – одна з фаз тестування програмного забезпечення, при якій окремі програмні модулі об'єднуються і тестуються в групі.

Регресійне тестування – вид тестування, спрямований на перевірку змін, зроблених в додатку, для підтвердження того факту, що працює раніше код працює, як і раніше [29].

5.1 Класифікація видів тестування програмного забезпечення

Існує кілька ознак, за якими прийнято проводити класифікацію видів тестування. Зазвичай виділяють наступні:

Структурне тестування:

- тестування чорного ящика;
- тестування білого ящика (біла коробка);
- тестування сірого ящика (сіра коробка).

По об'єкту тестування:

- функціональне тестування;
- тестування продуктивності;
- навантажувальне тестування;

- стрес тестування;
- тестування стабільності;
- юзабіліті-тестування;
- тестування інтерфейсу користувача;
- тестування безпеки служби (тестування безпеки);
- тестування локалізації ;
- тестування сумісності.

За ступенем автоматизації:

- ручне тестування;
- автоматизоване тестування;
- напіваавтоматичне тестування.

За ступенем ізольованості компонентів:

- компонентне (модульну) тестування;
- інтеграційне тестування;
- системне тестування.

За часом проведення тестування:

- альфа-тестування;
- тестування при прийманні (тестування диму);
- тестування нової функціональності;
- тестування при здачі (приймальні випробування);
- бета-тестування (бета-тестування).

За ознакою позитивності сценаріїв:

- позитивне тестування;
- негативне тестування.

За ступенем підготовленості до тестування:

- тестування по документації (формальне тестування);
- тестування спеціальних або інтуїтивне тестування (спеціальна перевірка).

5.2 Тестування основних функціональних можливостей

У наступних таблицях наведений перелік випробувань основних функціональних можливостей (таблиці 5.1, 5.2, 5.3, 5.4).

Таблиця 5.1 – Перевірка функції «Додати ЕЦП» при надсиланні документа

Мета тесту:	Перевірка функції «Додати ЕЦП» при надсиланні документа
Початковий стан системи	Відкритий веб додаток
Вхідні данні:	Шлях до особистого ключа користувача, що підписує документ та пароль до ключа
Схема проведення тесту:	Натиснути на кнопку «Додати ЕЦП» головної панелі Вказати шлях до приватного ключа та ввести пароль
Очікуваний результат:	ЕЦП успішно генерується та прикріплюється до оригіналів документу
Стан системи після проведення випробувань:	ЕЦП успішно генерується та прикріплюється до оригіналів документу

Таблиця 5.2 – Перевірка функції «Надіслати документ», коли документ не змінювався

Мета тесту:	Перевірка функції «Надіслати документ», коли документ не змінювався
Початковий стан системи	Відкритий веб додаток. Документ підписано за допомогою ЕЦП. Документ не змінювався після підписання
Вхідні данні:	

Схема проведення тесту:	Натиснути на кнопку «Надіслати документ».
Очікуваний результат:	Відображається повідомлення про те, що документ вдало надіслано.
Стан системи після проведення випробувань:	Відображається повідомлення про те, що документ вдало надіслано.

Таблиця 5.3 – Перевірка функції «Надіслати документ», коли документ змінювався

Мета тесту:	Перевірка функції «Надіслати документ», коли документ змінювався
Початковий стан системи	Відкритий веб додаток. Документ підписано за допомогою ЕЦП. Документ змінювався після підписання (наприклад, додали рядок)
Вхідні данні:	
Схема проведення тесту:	Натиснути на кнопку «Надіслати документ».
Очікуваний результат:	Відображається повідомлення про те, що документ не вдалося надіслати.
Стан системи після проведення випробувань:	Відображається повідомлення про те, що документ не вдалося надіслати.

Таблиця 5.4 – Перевірка надсилання документу

Мета тесту:	Перевірка надсилання документу
Початковий стан системи	Контрагенту Х надіслано електронний документ

Вхідні данні:	
Схема проведення тесту:	Зайти в веб додаток з ЕЦП контрагента Х. Знайди папку «Вхідні»
Очікуваний результат:	Відображається новий документ, що був надісланий контрагенту Х.
Стан системи після проведення випробувань:	Відображається новий документ, що був надісланий контрагенту Х.

5.3 Реалізація автоматичного тестування

Функціонал автоматичного тестування ПЗ в ERP системі, в тому числі в і в модулі особистого електронного кабінету базується на підходах інтеграційного, позитивного і регресійного тестування.

Інтеграційне тестування в якості вхідних даних використовує модулі, над якими було проведено модульне тестування, групує їх в більш великі безлічі, виконує тести, певні в плані тестування для цих множин, і представляє їх в якості вихідних даних і вхідних для подальшого системного тестування. Метою інтеграційного тестування є перевірка відповідності проєктованих одиниць функціональним, прийомним і вимогам надійності. Тестування ця проєктована одиниці - об'єднання, безліч або група модулі - виконується через їх інтерфейс, з використанням тестування «чорний ящик».

Позитивне тестування: якщо на вхід подати заздалегідь підготовлені вибірки вхідних даних і контрольного результату, то отриманий в процесі результат розрахунку повинен збігатися з контрольним.

Регресійне тестування передбачає перевірку змін, зроблених в додатку, для підтвердження того факту, що працює раніше код працює, як і раніше. У зв'язку з тим, що зміни в програмні забезпечення можуть вносити різні розробники, виникає необхідність в регулярному застосуванні цього підходу [30].

Будь-яка технологія тестування не дозволяє відловити всі помилки програми. Але за прийнятний час можна виконати перевірку спільної працездатності групи модулів на заздалегідь підготовлених вибірках даних і налаштувань. Для практичного використання в системі реалізовані механізми підготовка, зберігання, регламентний автоматичне виконання тестів і розсилка звітів про невідповідності і виконання тестування.

6 СТАРТАП-ПРОЕКТ

6.1 Опис ідеї проекту (товару, послуги, технології)

Розробка відноситься систем обміну юридично значущими електронними документами між контрагентами. Реалізація через хмарний сервіс та загальнодоступний портал вирішує проблеми термінальних рішень, такі як: установка на персональний комп'ютер та проведення оновлень системи.

На даний момент головною є багато областей застосування систем обміну юридично значущими електронним документами, наприклад, обмін документами між організаторами та постачальниками в електронних тендерних площадках. В таблиці 6.1 описано ідею стартап-проекту.

Таблиця 6.1 — Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Зменшення часу на обмін юридично значущими документами	1. Операції між контрагентами, які потребують обміну юридично значущими документами	Зменшення часу та витрат на обмін документами між користувачами

Відсутність потреби установки додатку на персональний комп'ютер та проведення оновлень, оплата тільки по факту використання сервісу робить розробку конкурентоспроможною. В таблиці 6.2 сильні, слабкі та нейтральні характеристики ідеї проекту.

Таблиця 6.2 — Визначення сильних, слабких та нейтральних характеристик ідеї проекту.

№ п/ п	Техніко-економічні характеристики ідеї	Мій проект	М.Е.Дос	Птах
1	2	3	4	5
1.	Вартість	Низька	Висока	Середня
2.	Термінальне рішення	-	+	-
3.	Швидкодія	Висока	Середня	Висока
4.	Хмарне рішення	+	-	+
5.	Відкрите арі	+	+	+

Продовження таблиці 6.2

№ п/ п	Техніко-економічні характеристики ідеї	W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
1	2	8	9	10
1.	Вартість			+
2.	Термінальне рішення			+
3.	Швидкодія			+
4.	Хмарне рішення			+
5	Відкрите арі			+

6.2 Технологічний аудит ідеї проекту

В таблиці 6.3 розглянуто технічну здійсненність ідеї проекту.

Таблиця 6.3 — Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Реалізація системи обмін юридично значущими електронними документами між контрагентами	Розміщення в хмарному середовищі, розробка веб-порталу з використанням технологій Asp.Net та Vue.js та використання ЕЦП для підписання документів.	Наявні.	Так.
Обраною технологією є Asp.Net та Vue.js для розробки порталу обміну документами та ЕЦП для підписання та надання документами юридичної значущості.				

6.3 Аналіз ринкових можливостей запуску стартап-проекту

Найголовнішими характеристиками даної продукції є розміщення системи в хмарному середовищі, невисока ціна, оплата тільки за фактичне користування сервісом та багатий функціонал. В таблиці 6.4 описано попередню характеристику потенційного ринку стартап-проекту.

Таблиця 6.4 — Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	2	3
1	Кількість головних гравців, од.	Закордонні та в Україні.

Продовження таблиці 6.4

1	2	3
2	Загальний обсяг продаж, грн/ ум.од.	Обмін одним документом в середньому коштує 1.5 грн/ ум.од.
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень).	Відсутні.
5	Специфічні вимоги до стандартизації та сертифікації.	Відсутні.
6	Середня норма рентабельності в галузі (по ринку), %	20

Основною споживчою аудиторією є різноманітні приватні фірми, підприємства, державні установи та інші. Система розповсюджується як сервіс та оплачується за фактичну кількість опублікованих документів. Розробка є універсальною. В таблиці 6.5 наведено характеристику потенційних клієнтів стартап-проекту.

Таблиця 6.5 — Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів
1	2	3	4
1. Невисока ціна		Зацікавлені в економії ресурсів.	Задовільна ціна, хороший функціонал

Продовження таблиці 6.5

1	2	3	4
2. Хмарне рішення	Збільшення операцій з необхідністю обміну документами між контрагентами	Зацікавлені в вищій якості, економії часу та ресурсів.	
3. Відкрите арі		Зацікавлені в інтеграції з іншими програмними рішеннями.	

Серед основних загроз є конкурентоспроможність, тому необхідна реклама, щоб потенційні користувачі дізналися про наявність даного продукту. В таблиці 6.6 розглянуто фактори загроз.

Таблиця 6.6 — Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	2	3	4
1.	Конкурентоспроможність	Невідомий бренд, нова система на ринку, низька конкурентоспроможність	Реклама, вдалий маркетинговий проект, залучення дилерів, спонсорів до співпраці.

Основною можливістю є залучення іноземних спонсорів та можливість реалізації товару закордоном. В таблиці 6.7 розглянуто фактори можливостей.

Таблиця 6.7 — Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Залучення партнерів	Залучення відомих компаній.	Укладення договорів про співпрацю та налагодження торгових контактів.
2.	Співпраця з державними установами	Залучення великого ринку з високими показниками документообігу.	Укладення договорів про співпрацю та підтримка на державному рівні.

В таблиці 6.8 проведено ступеневий аналіз конкуренції на ринку.

Таблиця 6.8 — Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції	Чиста.	Нижча ціна та кращий сервіс
2. За рівнем конкурентної боротьби	Національний	Нижча ціна та кращий сервіс
3. За галузевою ознакою	Внутрішньогалузева	Більший набір функцій.
4. Конкуренція за видами товарів	Товарно-родова	Реклама.
5. За характером конкурентних переваг	Цінова, якісна	Нижча ціна ніж в конкурентів та краща функціональність.
6. За інтенсивністю	Не марочна	Покращення технологій

В таблиці 6.9 проведено більш детальний аналіз умов конкуренції в галузі (за моделлю 5 сил М. Портера).

Таблиця 6.9 — Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Поста-чальники	Клієнти	Товари-замінники
Складові аналізу	Невелика кількість	Гнучкі ціни, розмір капітало-вкладень	Не потрібні	Якісна продукція	Вища ціна, вищі витрати, менше функцій
Висновки	Відносно невисока конкурентна боротьба	Присутні потенційні конкуренти	Не диктують	Так, залежно від попиту на товар.	Обмежень немає

На ринку не велика кількість конкурентів та високий попит. Конкурентна боротьба поки невисока, але з часом буде збільшуватись.

В таблиці 6.10 обґрунтовано фактори конкурентоспроможності.

Таблиця 6.10 — Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Функціональність	Висока, більша ніж в конкурентів
2	Ціна	Ґрунтовна на собівартості розробки, тому є нижчою.
3	Якість	На рівні конкурентів

В таблиці 6.11 проведено Порівняльний аналіз сильних та слабких сторін власного проекту.

Таблиця 6.11 — Порівняльний аналіз сильних та слабких сторін власного проекту

№ п/ п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з власною компанією						
			-3	-2	-1	0	+1	+2	+3
1	Функціональність	20		+					
2	Ціна	17		+					
3	Якість	14						+	

В таблиці 6.12 проведено SWOT-аналіз стартап-проекту.

Таблиця 6.12 — SWOT-аналіз стартап-проекту

Сильні сторони: Ціна основана на вартості розробки Рентабельність Відкрите арі	Слабкі сторони: Новий, невідомий продукт
Можливості: Забезпечення споживчих потреб Високий дохід	Загрози: Недостатньо досконала реалізація

В таблиці 6.13 розглянуто альтернативи ринкового впровадження стартап-проекту.

Таблиця 6.13 — Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Практичне використання та вдосконалення системи	+	1 рік
3	Рекламна кампанія	+	0,5 року
5	Вихід на закордонний ринок	+	2 роки

Обраною альтернативою є залучення рекламної кампанії, розкрутка нової системи.

6.4 Розроблення ринкової стратегії проекту

В таблиці 6.14 проведено вибір цільових груп потенційних споживачів.

Таблиця 6.14 — Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Привітні компанії	Готові	Високий	Велика	Середня
2	Державні установи	Готові	Високий	Велика	Середня
Які цільові групи обрано: приватні компанії.					

В таблиці 6.15 проведено визначення базової стратегії розвитку.

Таблиця 6.15 — Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкуренто- спроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	2	3	4	5
1	Приватні компанії	Стратегія лідерства по витратах	<ul style="list-style-type: none"> – низька ціна – простота в користуванні – продуктивність 	Контроль постійних витрат, зниження виробничих, реklamних і збутових витрат, проведення інвестування, спрямованого на зменшення витрат

В таблиці 6.16 визначено базової стратегії конкурентної поведінки

Таблиця 6.16 — Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Буде шукати	Так, основний функціонал у всіх конкурентів схожий	Стратегія наслі- дування лідера

В таблиці 6.17 визначено стратегії позиціонування.

Таблиця 6.17 — Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	2	3	4	5
1	Низька ціна.	Стратегія наслідування лідера	<ul style="list-style-type: none"> — низька ціна — низькі витрати на сировину — простота виготовлення — велика кількість виготовленої продукції 	<ul style="list-style-type: none"> — зниження ціни — універсальність для кожного клієнта — кращий функціонал
	Відкрите арі			
	Простота використання			

6.5 Розроблення маркетингової програми стартап-проекту

В таблиці 6.18 Визначено ключові переваги концепції потенційного товару.

Таблиця 6.18 — Визначення ключових переваг концепції потенційного товару

№п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	2	3	4
1	Дешевизна	Низька ціна	Ціна

Продовження таблиці 6.18

1	2	3	4
2	Відкрите арі	Інтеграція з будь-якими системами	Універсальність
3	Простота використання	Користувачу легше виконувати свої задачі	Функціональність

В таблиці 6.19 описано трьохрівневу модель товару.

Таблиця 6.19 — Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Система обміну юридично значущими електронними документами між контрагентами		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Простота використання		
	2. Широкий функціонал		
	Якість: стандарти, нормативи, параметри тестування тощо Стандартизація відповідно до ГОСТ, ISO, МЕК. Регламентується ГОСТ, ISO, МЕК.		
	Пакування відсутнє.		
	Марка: назва організації-розробника + назва системи		
Потенційний товар буде захищено від копіювання: патентування, сертифікати відповідності.			

В таблиці 6.20 проведено визначення меж встановлення ціни.

Таблиця 6.20 — Визначення меж встановлення ціни (із розрахунку на 1 рік користування)

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
>9000 грн.	8000-9000грн.	10-20 тис. грн	5300-6500грн

В таблиці 6.21 розглянуто формування системи збуту.

Таблиця 6.21 — Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Спочатку ознайомлення з функціоналом	Відсутні	Канал 0-го рівня	Власний веб портал

В таблиці 6.22 розглянуто концепцію маркетингових комунікацій.

Таблиця 6.22 — Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Довіра до приватних компаній - партнерів	Обмін юридичними документами	Налагодження контактів приватними компаніями	Донести ідею, інформацію про функції	Продемонструвати переваги

6.5 Висновки

Ідеєю стартапу є підвищення ефективності обміну документами між контрагентами через розробку загальнодоступної системи обміну документами в електронному вигляді з підтримкою ЕЦП.

Є можливість ринкової комерціалізації проекту, так як наявний попит та потреба у даній системі на ринку України.

Чиста конкуренція, локальна внутрішньовидова конкуренція, що сприяє для успішного входження на ринок.

Альтернативою можна назвати низьку ціну та постійну технічну підтримку. Подальша імплементація є доцільною для даного проекту.

ВИСНОВКИ

У дипломному проекті був розроблений програмний комплекс «Система обміну юридично значущими електронними документами між контрагентами».

Дане рішення відкриває має такі можливості: для роботи лише браузер та вихід в Інтернет, єдиний логін в інфраструктурі сервісів від IT-Enterprise, вхід по ЕЦП, автоматична реєстрація, створення файлової структури для користувача чи групи, завантаження/збереження файлів на сервіс в тому числі з підписами, скачування файлів з сервісу в тому числі з підписами, перейменування та переміщення файлів на сервісі, робота з версіями документів, підписання документів, перегляд існуючих підписів, передача документів на підпис іншому користувачу, надання доступу/публікація іншим користувачам, визначення операцій надається доступ: перегляд, редагування, підпис, публікація.

Для реалізації даного рішення було обрано сервісно-орієнтовану архітектуру. Веб сервіси побудовані на основі технології ASMX, яка входить до платформи .NET і підтримує протокол SOAP.

Сайт порталу побудований основі технології ASP.NET MVC. ASP.NET - це технологія створення веб-застосунків і веб-сервісів від компанії Майкрософт. ASP.NET має перевагу у швидкості в порівнянні з іншими технологіями, заснованими на скриптах (PHP, тощо). Розширюваний набір елементів управління і бібліотек класів дозволяє швидко розробляти застосунки.

Шаблон MVC дозволяє створювати додатки, різні аспекти яких (логіка введення, бізнес-логіка і логіка інтерфейсу) розділені, але досить тісно взаємодіють один з одним.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Конституція України (зі змінами): Прийнята на п'ятій сесії Верховної Ради України 28 червня 1996 року // Голос України. – 1996.
2. Про платіжні системи та переказ коштів в Україні (зі змінами та доповненнями). Закон України від 5 квітня 2001 р. // ВВР. – 2001. – № 29. – Ст. 137.
3. Про електронний цифровий підпис (зі змінами та доповненнями). Закон України від 22 травня 2003 р. // ВВР. – 2003. – № 36. – Ст. 276.
4. Про електронні документи та електронний документообіг (зі змінами та доповненнями). Закон України від 22 травня 2003 р. // ВВР. – 2003. – № 36. – Ст. 275.
5. Азаров Д.С. Злочини у сфері комп'ютерної інформації (кримінально-правове дослідження): Монографія. – К. : Атака, 2007. – 304 с.
6. Захарченко В.Ю., Лазуренко В.И., Олифіров А.В., Рогозин С.Н. Компьютерные преступления: их выявление и предотвращение: Учебное пособие / Под общ. ред. В.И. Лазуренко. – К. : Центр учебной литературы. – 2007. – 170 с.
7. Мотлях О.І. Методика розслідування комп'ютерних злочинів. Монографія. – К. : Освіта України, 2010. – 296 с.
8. Шаньгин В.Ф. Информационная безопасность компьютерных систем и сетей : учеб. пособие, М.: «ФОРУМ»: ИНФРА-М, 2008. – 416 с.
9. Johnson D. and Vanstone S. The elliptic curve digital signature algorithm (ECDSA) // International Journal on Information Security, 1 (2001). – Pp. 36-63.
10. Алгулиев Р.М., Имамвердиев Я.Н.. Исследование международных и национальных стандартов цифровой подписи на эллиптических кривых // Вопросы защиты информации. – Москва, 2005. – №2(69).

11. Д.Н. Молдовян. Новый механизм формирования подписи в схемах ЭЦП, основанных на сложности дискретного логарифмирования и факторизации // Вопросы защиты информации. – Москва, 2005. – №4 (71).

12. Венбо Мао, Современная криптография. Теория и практика. – Изд.:Лори Вильямс, 2005.

13. Б. Шнайер. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – Москва: ТРИУМФ, 2002 – 816 с.

14. Д.Е. Акбаров. Криптография, стандарты алгоритмов криптографической защиты информации и их приложения. – Ташкент, 2007 г. – 188 с.

15. Элементарное введение в эллиптическую криптографию: алгебраические и алгоритмические основы / Болотов А.А. Гашков С.Б. Фролов А.В., Часовских А.А. – Москва МЭИ, 2006. – 328 с.

16. "Методы и средства защиты информации" (курс лекций) Беляев А.В.

17. И.Д. Горбенко, С.И. Збитнев, А.А. Поляков Криптографические преобразования в группах точек эллиптических кривых методом Полларда // Радиотехника: Всеукр. межвед. науч-тех. сб 2001. Вып. 119. С. 43-50.

18. И.Д. ГОРБЕНКО, д-р техн. наук, А.А. ПОЛЯКОВ, С.И. ЗБИТНЕВ, ПРОТОКОЛЫ – ПРИМИТИВЫ УПРАВЛЕНИЯ КЛЮЧАМИ В ГРУППАХ ТОЧЕК ЭЛЛИПТИЧЕСКИХ КРИВЫХ

21. ERP-системы: выбор, внедрение, эксплуатация. Современное планирование и управление ресурсами предприятия / Дэниел О’Лири - М.: Вершина, 2004 – 272с.

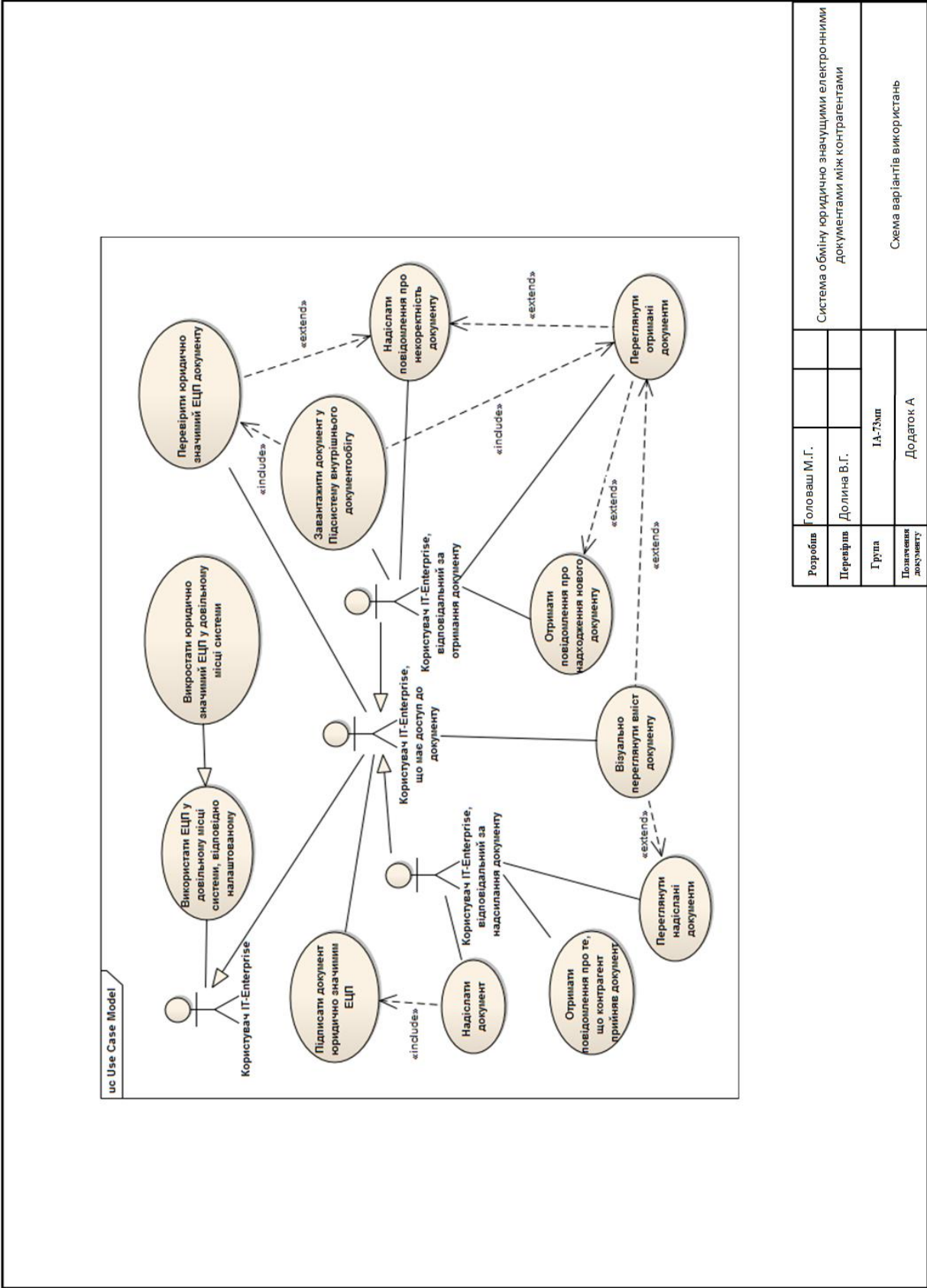
23. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер — К.: ООО Издательство «Питер», 2013. — 896 с.

24. Троелсен, А. Мова програмування C# 5.0 і платформа .NET 4.5 / Дж. Рихтер — К.: ООО «И.Д. Вильямс», 2013. — 1310 с.

25. Мартін Р. С., Мартин М. Принципи, патерни і методики гнучкої розробки на мові С #; Символ-Плюс, 2011. - 768 с.
26. Фаулер, М. Улучшение существующего кода / М. Фаулер — К.: ООО Издательство «Символ-Плюс», 2013. — 708 с.
27. Мартин, Гр. Введение в SQL / Гр. Мартин — К.: Лори, 2010. — 227 с.
28. Буч, Гр. Язык UML. Руководство пользователя / Гр. Буч. — Москва: ДМК Пресс, 2007. — 200 с.
29. Лайза Криспин, Джанет Грегори Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд = Agile Testing: A Practical Guide for Testers and Agile Teams. — М. : «Вильямс», 2010. — 464 с.
30. Калбертсон Роберт, Браун Крис, Кобб Гэри Быстрое тестирование. — М. : «Вильямс», 2002. — 374 с.
31. Бейзер Б. Тестирование чёрного ящика. Технологии функционального тестирования программного обеспечения и систем. — СПб. : Питер, 2004. — 320 с.

ДОДАТОК А

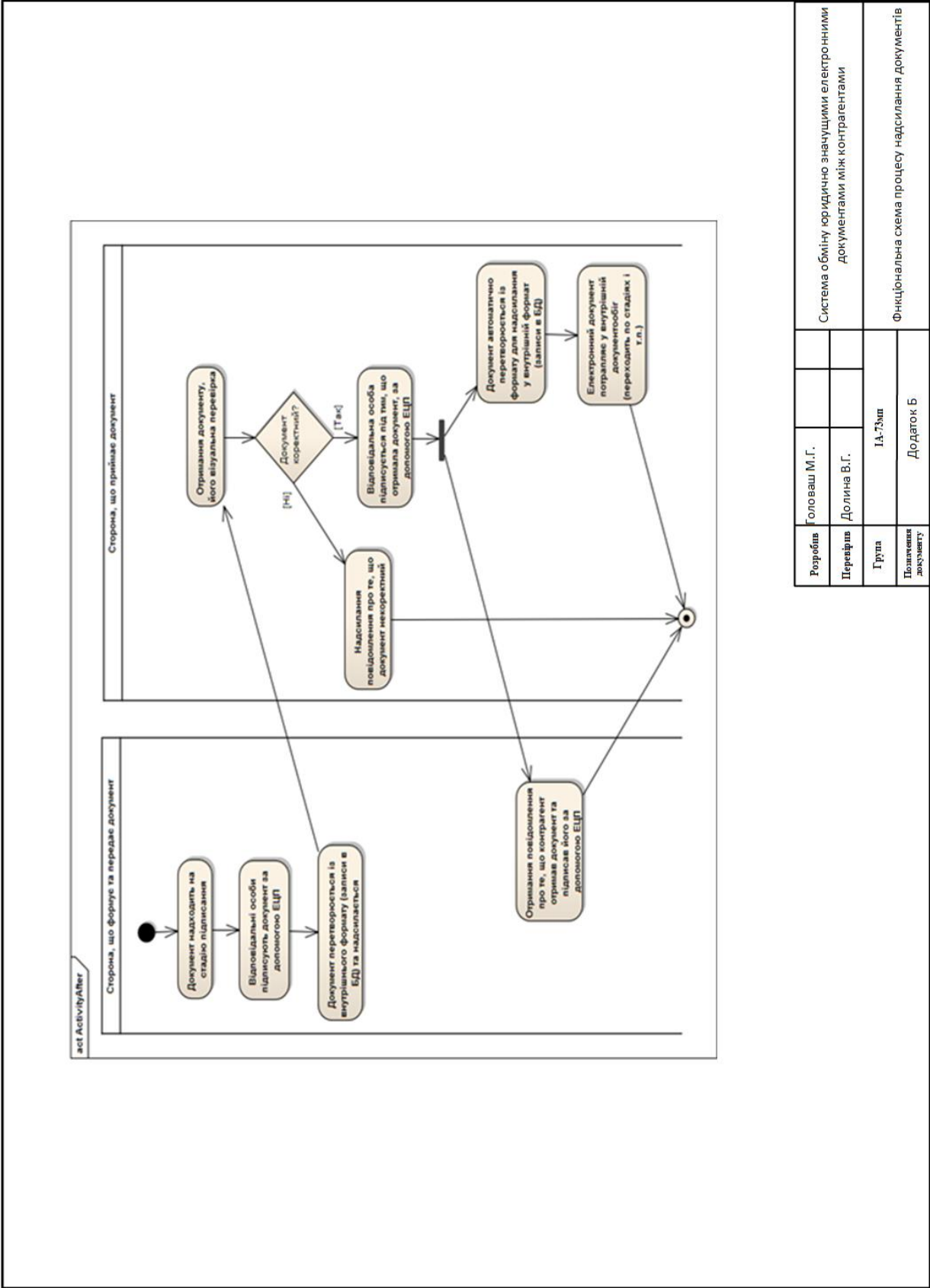
Система обміну юридично значущими електронними документами між контрагентами. Схема варіантів використання



Розробка	Головаш М.Г.				Система обміну юридично значущими електронними документами між контрагентами
Перевірка	Долина В.Г.				
Група	ІА-7мп				Схема варіантів використання
Позначення документа	Додаток А				

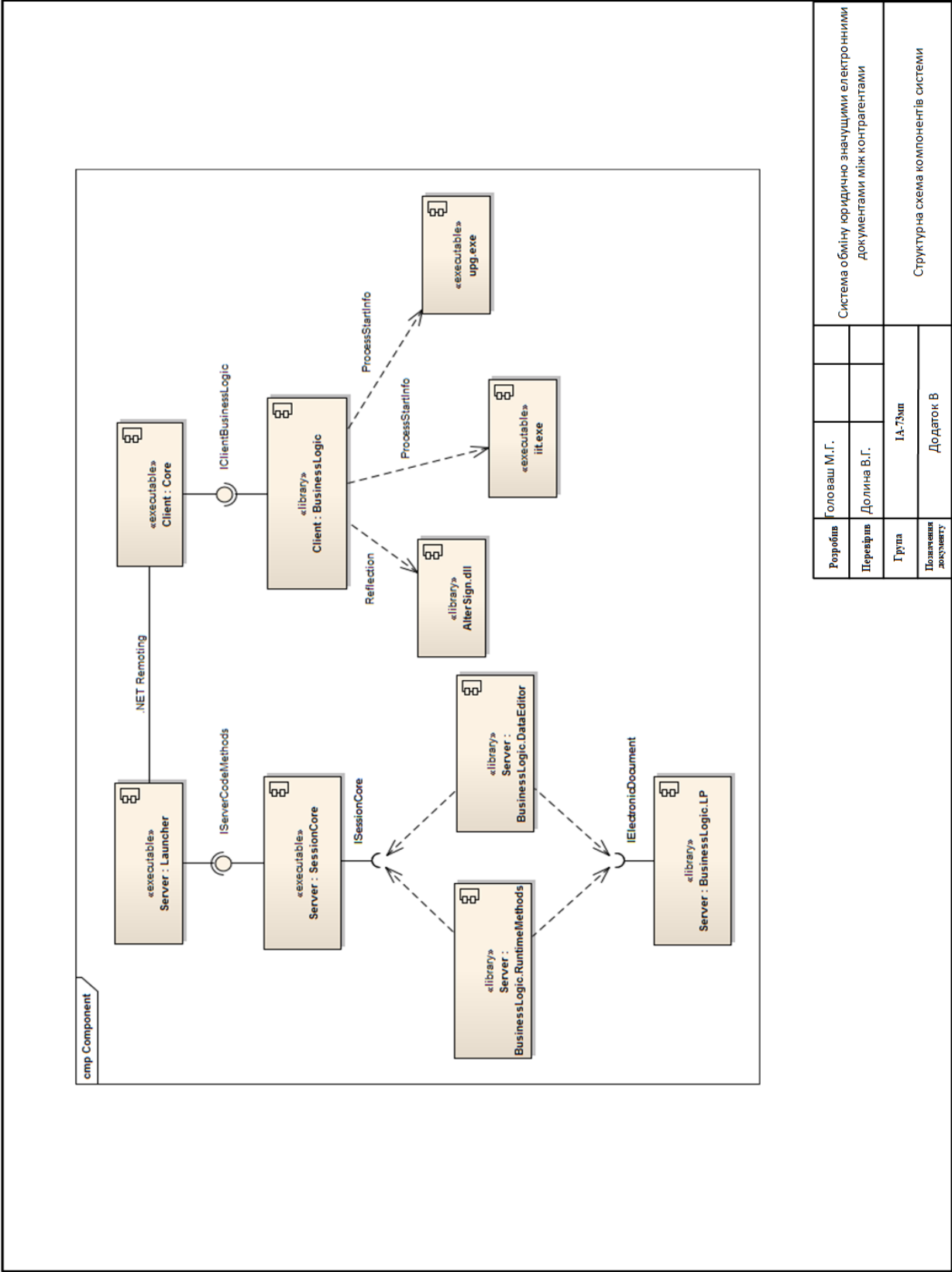
ДОДАТОК Б

Система обміну юридично значущими електронними документами між контрагентами. Функціональна схема процесу надсилання документів



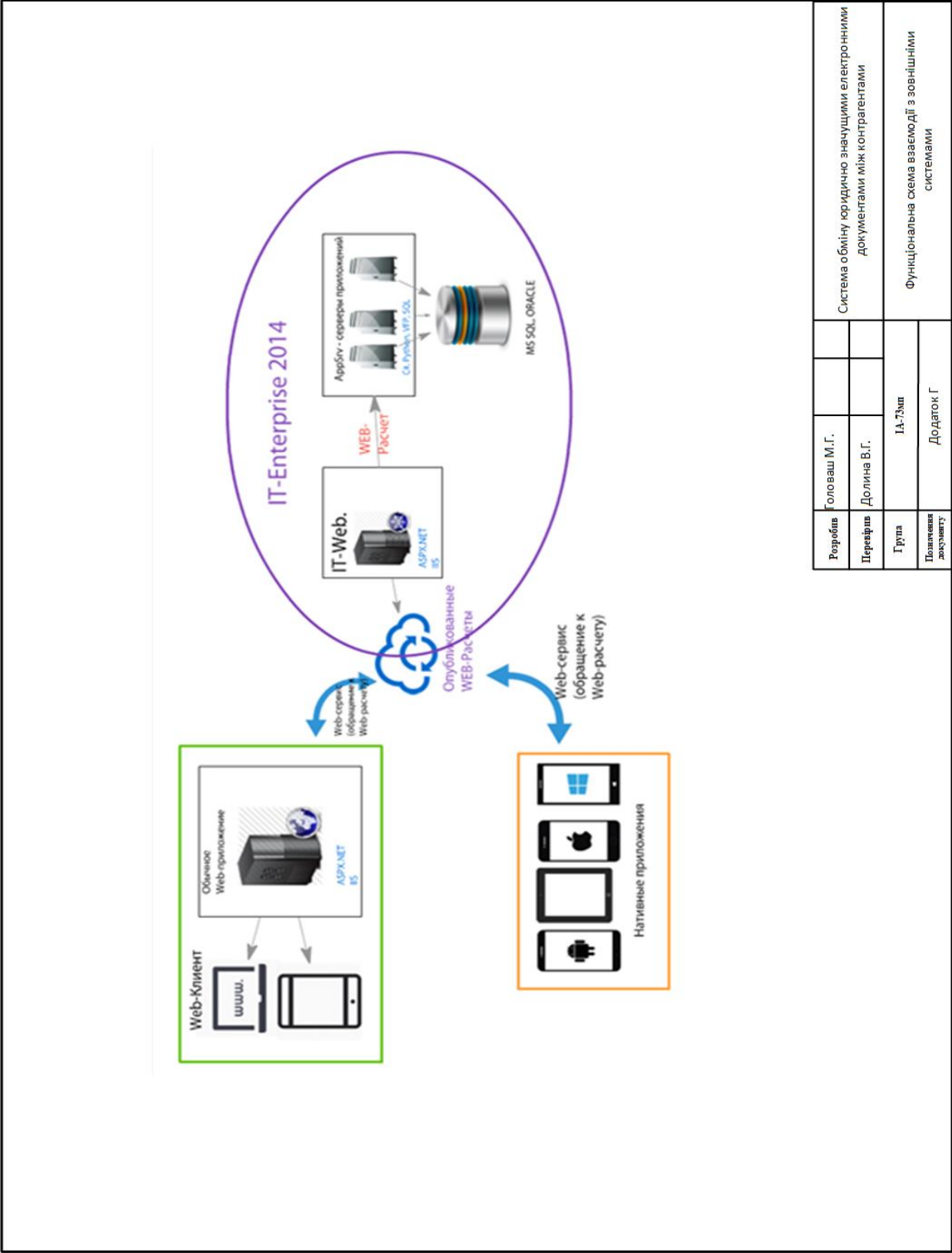
ДОДАТОК В

Система обміну юридично значущими електронними документами між контрагентами. Структурна схема компонентів системи



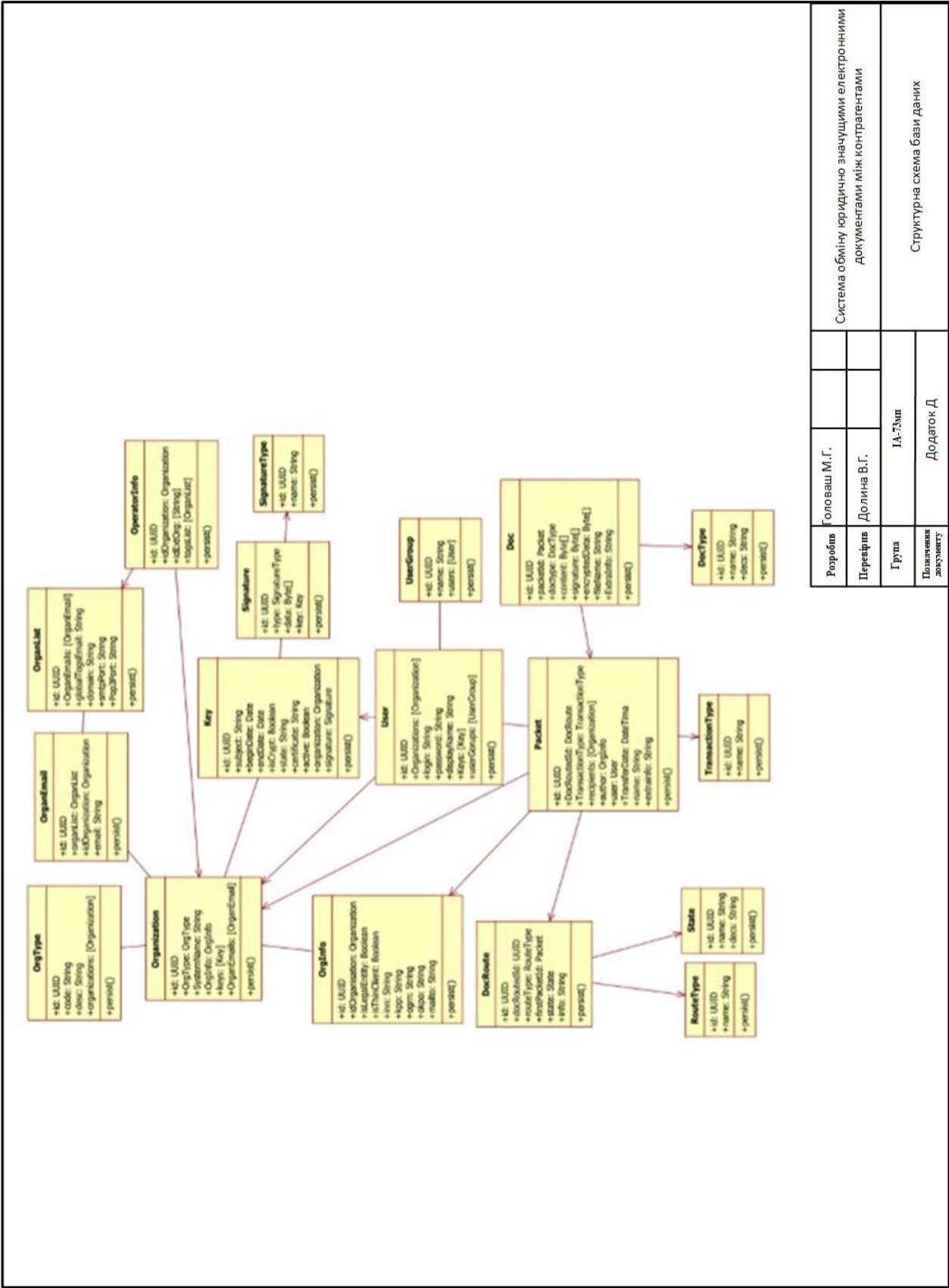
ДОДАТОК Г

Система обміну юридично значущими електронними документами між контрагентами. Функціональна схема взаємодії із зовнішніми системами



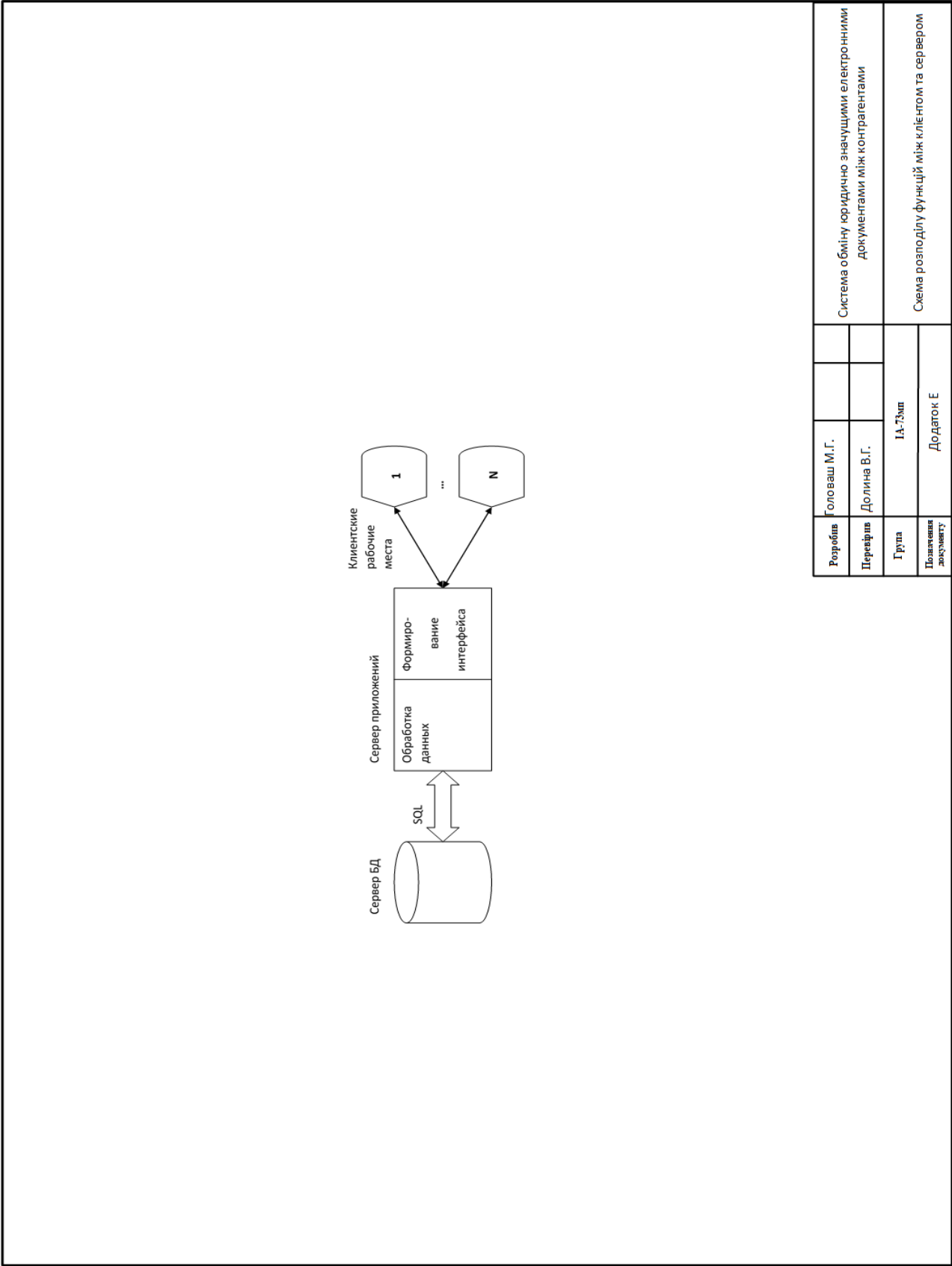
ДОДАТОК Д

Система обміну юридично значущими електронними документами між контрагентами. Структурна схема бази даних



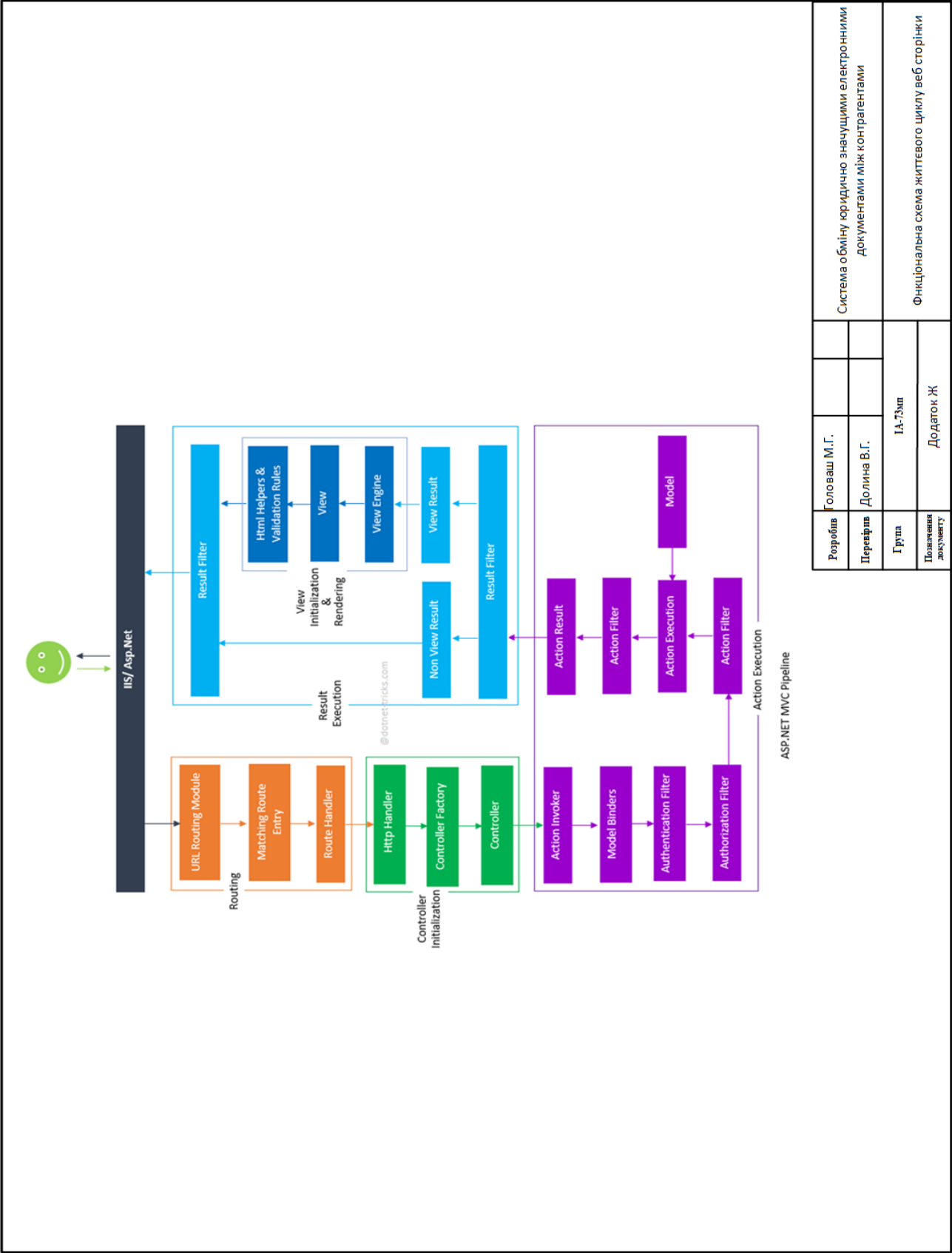
ДОДАТОК Е

Система обміну юридично значущими електронними документами між контрагентами. Схема розподілу функцій між клієнтом та сервером



ДОДАТОК Ж

Система обміну юридично значущими електронними документами між контрагентами. Функціональна схема життєвого циклу веб сторінки



ДОДАТОК 3

Система обміну юридично значущими електронними документами між контрагентами. Креслення вигляду сторінки веб порталу

Розробив	Головаш М.Г.			Система обміну юридично значущими електронними документами між контрагентами
Перевірив	Долина В.Г.			
Група	ІА-73м			
Положення документу	Додаток 3			Креслення вигляду сторінки веб порталу